



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PODPORA HRY KRYCÍ JMÉNA NA MOBILNÍM TELEFONU S OS ANDROID

SUPPORT FOR CODENAMES GAME ON MOBILE PHONE WITH OS ANDROID

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN GROSSMANN

VEDOUCÍ PRÁCE

SUPERVISOR

Doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Grossmann Jan**

Obor: Informační technologie

Téma: **Podpora hry Krycí jména na mobilním telefonu s OS Android**
Support for Codenames Game on Mobile Phone with OS Android

Kategorie: Počítačová grafika

Pokyny:

1. Seznamte se s dosavadními jednoduchými aplikacemi pro podporu hry Krycí jména a metodami pro zpracování a rozpoznávání obrazu v OS Android.
2. Shromážděte datovou sadu snímků herního plánu Krycích jmen v různých fázích hry a připravte data pro průběžné vyhodnocování kvality rozpoznávání a kontroly konzistence.
3. Navrhněte a implementujte systém pro usnadnění hraní Krycích jmen pomocí rozpoznávání a zpracování na mobilním telefonu.
4. Vyhodnoťte realizované řešení v adekvátní uživatelské studii v reálném prostředí při hraní.
5. Vytvořte stručný plakát prezentující práci, její cíle a výsledky.

Literatura:

- dle doporučení vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- funkční prototyp řešení

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese
<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Smrž Pavel, doc. RNDr., Ph.D.,** UPGM FIT VUT

Datum zadání: 1. listopadu 2017

Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato bakalářská práce se zabývá tvorbou aplikace pro podporu hraní společenské hry Krycí jména pro zařízení s operačním systémem Android. Aplikace pomáhá uživateli se strategií a usnadňuje rozhodování při podávání nápovědy. V práci nejprve probírám stávající řešení a jejich nedostatky. Na základě těchto poznatků rozebírám navržené řešení a poté i samotnou implementaci pomocí programovacího jazyka Java, do které patří uložení dat pomocí databázového systému nebo také rozpoznávání obrazu. Závěrem provádím a detailně popisuji uživatelské testování a důležité poznatky z něho vyplývající.

Abstract

This bachelor's thesis deals with creation of an application for support for the Codenames game on mobile phone with Android operating system. Application helps user with game strategy and simplify selection of the clue. First I discuss existing solutions and their imperfections. Based on this experience, I analyze designed solution and then, the very implementation with usage of Java programming language, involving storing data with database system or optical recognition. Finally, I undertake user testing, which I also describe in detail.

Klíčová slova

Android aplikace, mobilní aplikace, chytré zařízení, Java, Google, OpenCV, Krycí jména, rozpoznávání obrazu, Python, SQL, SQLite

Keywords

Android application, mobile application, smart device, Java, Google, OpenCV, Codenames Game, optical recognition, SQL, Python, SQLite

Citace

GROSSMANN, Jan. *Podpora hry Krycí jména na mobilním telefonu s OS Android*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. RNDr. Pavel Smrž, Ph.D.

Podpora hry Krycí jména na mobilním telefonu s OS Android

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. RNDr. Pavla Smrže, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jan Grossmann

16. května 2018

Poděkování

Rád bych poděkoval svému vedoucímu práce za cenné rady a podněty.

Obsah

1	Úvod	2
2	Společenská hra Krycí jména	3
2.1	Obsah herního balení	3
2.2	Prerekvizity	3
2.3	Průběh hry	3
2.4	Herní variace	4
3	Podobné aplikace	5
3.1	OS Android	5
3.2	Další platformy	6
3.3	Podpora her na mobilním telefonu	7
4	Moje Aplikace	9
4.1	Popis	9
4.2	Průběh hry	9
4.3	Pomoc	13
4.4	.CNR záznam	15
5	Implementace	16
5.1	Vývoj pro OS Android	16
5.2	Základní kameny Java Android aplikací	18
5.3	Uchování a získání nápověd	20
5.4	Rozpoznávání	23
5.5	Backend	28
6	Testování a experimenty	31
6.1	Intuitivnost rozhraní	31
6.2	Úspěšnost rozpoznávání	33
6.3	Vliv na výhru	33
6.4	Vyhodnocení	35
7	Závěr	37
	Literatura	38
A	Sada snímků v různých fázích hry	39
B	Návod pro spuštění podpůrného serveru	42

Kapitola 1

Úvod

Chytrá mobilní zařízení se dnes stala již permanentním společníkem člověka. Lidé bez nich neopouští domov, v jejich blízkosti usínají a následně se probouzí, nosí je s sebou po domě, ba dokonce i na toaletu. Tohoto mnohdy až nezdravého vztahu lze využít všelijak, v mém případě jsem využil silné nezkrotné touhy po vítězství, kterou je jistě obdařen každý jedinec již od narození. Kdo nikdy nezneužil nepozornosti soupeřů a neposunul se o políčko dopředu, nebo se nepovoleně neobohatil v dostizích a sázkách. V trochu jiném provedení, ale stále se stejnou myšlenkou je tvořena i tato aplikace. Na rozdíl od stávajících aplikací v obchodě Google Play doslova podává pomocnou ruku hráči při hraní vysoce kompetitivní hry Krycí jména, ať už při napovídání slov ze slovníku, které mají nejvyšší potenciál, nebo při odhalování zamýšlených karet, díky funkci skenu je možno celý průběh hry monitorovat v chytrém zařízení a následně jej sdílet.

Tato práce si klade za cíl návrh a vytvoření nejen aplikace, usnadňující týmu průchod hrou za použití chytrého mobilního zařízení s operačním systémem Android, ale také vytvoření jednoduchého podpůrného serveru, na kterém probíhá rozšiřování slovní zásoby aplikace, a se kterým dokáže aplikace komunikovat v okamžiku potřeby sdílení dat. Dalším benefitem aplikace je, že v případě nedostatku hráčů může aplikace zastat roli jednoho z hlavních agentů, kdy osoba mající přehled o rozložení hry předčítá soupeřovu týmu slova objevující se na displeji.

V průběhu tvorby práce se zabývám programováním uživatelského rozhraní pro OS Android, rozpoznáváním obrazu technologií OpenCV při načítání herních plánů, rozpoznávání textu při načítání hracích karet, uložení dat do databázového souboru pomocí SQLite a tvorbou slovníku nápověd, komunikací se serverem pomocí REST API.

Kapitola 2

Společenská hra Krycí jména

Krycí jména je společenská hra, ve které soupeří dva týmy hráčů (červení a modří), kdy minimální počet hráčů v týmu je 2 a maximální 4. Hra využívá jednoduché, ale přesto zábavné mechaniky a proto se velmi rychle stala velice populární¹. Hra je stále svižná, průměrné kolo trvá asi 15-20 minut a neustále nutí všechny účastníky zůstat ve střehu.

2.1 Obsah herního balení

Herní balení zobrazené na obrázku 2.1 obsahuje 200 oboustranně potištěných herních karet, 40 herních plánů a 25 překrývacích karet. Na každé straně herních karet se nachází slovo - podstatné jméno. Z pětadvaceti překrývacích karet je 17 v barvách týmů, s pevným počtem 8 červených karet, 8 modrých karet a 1 oboustranná karta, která se v průběhu hry považuje za barvu, shodnou s barvou začínajícího týmu. Dále je zde i 7 karet ve žluté barvě, které znamenají náhodné kolemjdoucí a 1 černá karta reprezentující nájemného zabijáka. Herní plán vyobrazuje tabulku 5 krát 5 políček, barevně vyplněnou podle počtu jednotlivých překrývajících karet. Celkem tedy 4 barvy – červená, modrá, žlutá a černá. V neposlední řadě obsahuje balení i uživatelskou příručku a přesýpací hodiny.

2.2 Prerekvizity

Aby byla hra férová, je potřeba sudý počet hráčů. Hráči se rozdělí na 2 týmy, přičemž si každý tým zvolí svého kapitána, tzv. *Hlavního agenta*. Zbytek týmu se dále bude nazývat *operativci*. Na herní plochu se vyskládá 25 náhodných karet se slovy a uspořádají se do tvaru mřížky 5 krát 5 karet. K těmto kartám se náhodně vybere herní plán, který mají právo vidět pouze hlavní agenti a nikoliv operativci. Herní plán se na začátku kola může libovolně natočit a poté se umístí do stojanu kolmo k hrací ploše. Herní plán jednak určuje začínající tým, ale také příslušnost karet rozložených na stole k jednomu ze dvou soupeřících týmů, případně určuje pozici náhodných kolemjdoucích a také nájemného zabijáka.

2.3 Průběh hry

Úkolem hlavního agenta je podat operativcům náповědu, které karty přísluší jejich týmu. Bohužel ale může použít pouze jedno slovo a číslo značící počet zamýšlených karet repre-

¹Hra získala prestižní ocenění Spieles des Jahres 2016 – <http://www.spiel-des-jahres.com/de/codenames-ist-das-spiel-des-jahres-2016>



Obrázek 2.1: Obsah balení. zdroj: <http://www.mindok.cz/3/hry/detail/8595558302239-kryci-jmena-254>

zentovaných touto nápovědou. Operativci se postupně snaží nápovědy rozluštit a označit všechny karty jejich barvy, pokud označí kartu protihráče nebo náhodného kolemjdoucího, tah týmu končí a na řadu se dostává protivníkům tým se stejným úkolem. Pokud tým označí nájemného zabijáka, hra instantně skončí a soupeřům tým zvítězí. Hlavním cílem každého týmu je označit všechny jejich karty na hrací ploše dříve, než tak učiní soupeř.

2.4 Herní variace

Nejjednodušší rozšíření základního konceptu hry je přidání do hry přesýpacích hodin. Hodiny obsažené v balení se celé přesypou za asi 90 sekund, za tuto dobu musí hlavní agent vymyslet nápovědu pro tým a za další stejně dlouhý časový úsek se musí operativci pokusit uhádnout zamýšlená slova.

Výrobci hry Krycí jména přišli na trh s aktualizovanou variantou hry Krycí jména: Duet. Jedná se o hru dvou hráčů, s téměř totožnými herními principy, ale pozměněným herním plánem. Další variantou hry jsou Krycí jména: Pictures. V této modifikaci jsou karty s vytištěnými slovy nahrazeny obrázky.

Kapitola 3

Podobné aplikace

Hra Krycí jména se vyskytuje na celosvětovém trhu s deskovými hrami delší dobu, a proto již určitě existují podobné podpůrné aplikace. Průzkumem, jaké aplikace již existují a seznámením s nimi je možné odhalit nedostatky, nebo ještě před implementací posoudit co má opravdu smysl tvořit, a co je pouhá funkcionalita do počtu.

3.1 OS Android

Pro platformu OS Android se již vyskytuje několik aplikací zabývajících se tematikou hry Krycí jména. Většina z nich je v anglickém jazyce, podporuje generaci náhodného herního plánu a odpočtu času ovšem tyto funkce jsou převážně jen doplňující a k samotnému hraní hry nepřidávají téměř žádnou hodnotu.

3.1.1 Codenames Gadget

Jedná se o oficiální aplikaci¹ vydanou *CGE digital* graficky sladěnou s originálními motivy a kombinacemi barev vyskytující se na krabicovém balení hry. Aplikace podporuje pouze 2 funkce, a to vygenerování náhodného herního plánu a funkci odpočtu času, ale s tou výhodou že automaticky střídá odpočet pro hlavního agenta a operative, a to v obou týmech. Screenshot aplikace lze vidět na obrázku 3.1.

3.1.2 Codenames Helper

Podobný nástroj² jako oficiální aplikace, ovšem v minimalistickém designu (obrázek ??). Taktéž podporuje generování náhodného herního plánu a funkci odpočtu času. Vygenerovaný herní plán jde sdílet s kolegy pomocí čtyřmístného kódu, který si kdokoli může zadat ručně a zobrazit stejnou kombinaci.

3.1.3 Codenames with a Pictures Twist: The Curator

Varianta hry³ na způsob Krycí jména: Pictures od cizího vývojáře. Jedná se o mobilní multiplayerovou adaptaci s drobným rozšířením v podobě skryté bomby pod náhodnou kartu. Umožňuje hrát hru bez nutnosti zakoupení krabicové verze.

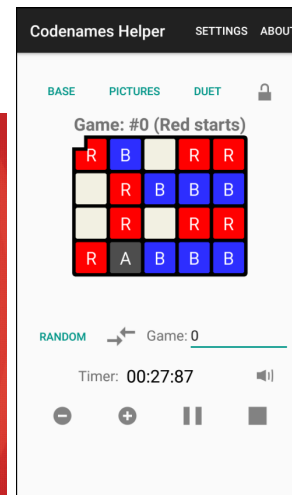
¹<https://play.google.com/store/apps/details?id=com.czechgames.codenames>

²<https://play.google.com/store/apps/details?id=com.effervex.codenameshelper2>

³<https://play.google.com/store/apps/details?id=com.groupfun.partyteamgame.curator>



(a) Codenames Gadget



(b) Codenames Helper

Obrázek 3.1: Oficiální aplikace Codenames Gadget podporující automatické střídání tahů a generování herního plánu a Codenames Helper pro odpočet času a generování náhodného herního plánu.

3.1.4 Codenames PnP

PnP v názvu znamená Print and Play. Pro hraní pomocí této aplikace⁴ je tedy zapotřebí vytisknout a vystříhat speciální kartičky obsahující čárový kód. Tyto karty pak lze jednu po druhé oskenovat a přehledně zobrazit seznamy karet. Aplikace cílí na hlavního agenta a pohodlí při výběru nápovědy, kdy zobrazuje pouze karty určité barvy.

3.1.5 Codenames Companion

Aplikace⁵ cílí na hlavního agenta, umožňující ručně zadat 8 nebo 9 slov k uhádnutí a následně si zaznamenávat kombinace nápověd a karet se kterými souvisí. Jako doplňující funkce je zde náhrada přesýpacích hodin.

3.2 Další platformy

3.2.1 Web HorsePaste.com

Na adrese HorsePaste.com⁶ běží webová aplikace, která umožní hrát hru Krycí jména online společně s přáteli. Hlavní koncept je vytvoření hry a následné zaslání odkazu všem účastníkům. Bohužel nejde rozlišit odkaz pro hlavní agenty od operativců a proto je otázka podvádění při hraní hry hlavně o důvěře. Po započetí hry se v okně prohlížeče zobrazí tabulka 5 * 5 karet, ukázka na obrázku 3.2. Stránka informuje hráče o tom, kdo je na tahu, kolik karet každému týmu zbývá a v reálném čase synchronizuje herní plochu mezi všemi přítomnými. Aplikace se zaměřuje pouze na samotné hraní hry bez nutnosti zakoupení kra-

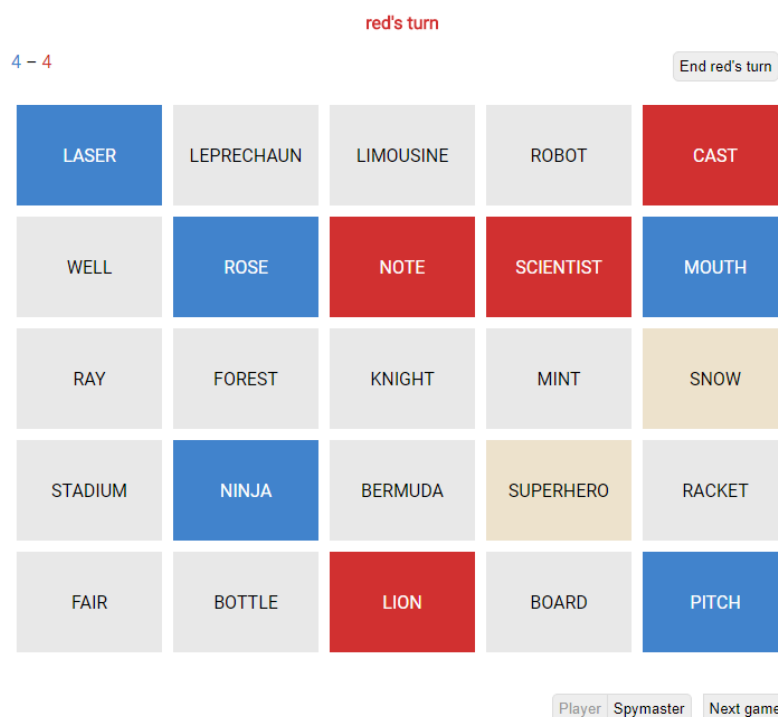
⁴<https://play.google.com/store/apps/details?id=com.hfd.codenames>

⁵<https://play.google.com/store/apps/details?id=com.tonykazanjan.codenamescompanion>

⁶<https://www.horsepaste.com>

CODENAMES

Send this link to friends: <https://www.horsepaste.com/overseas>



Obrázek 3.2: Aplikace pro hraní CodeNames na adrese HorsePaste.com

bicového balení, ovšem jedná se o skvělou alternativu pro hru s přáteli bez možnosti se osobně sejít. Jazyková sada karet je pouze v angličtině.

3.3 Podpora her na mobilním telefonu

Kromě podpory hry Krycí jména se na platformě OS Android vyskytuje velké množství aplikací pro podporu jiných her, jako jsou šachy, sudoku, křížovky, karty atd.

3.3.1 Chessify - Scan, Analyze, Play

Chessify⁷ je aplikace umožňující naskenování a digitalizaci šachových diagramů z knih, časopisů, pdf souborů nebo jiných zdrojů v libovolné fázi hry a následné vyřešení pomocí šachového engine Stockfish⁸, nebo možnost pokračovat ve hře proti počítači. Naskenované diagramy si lze ukládat. Jako doplňková funkce se zde nachází šachové hodiny.

⁷<https://play.google.com/store/apps/details?id=com.fimotech.chessfimee>

⁸<https://stockfishchess.org/>



Obrázek 3.3: Vyobrazení rozšířené reality z aplikace Sudoku Photo Solver. Obrázek převzatý z [6].

3.3.2 CrosScan

CrosScan, je aplikace popsaná v práci *CrosScan: The Crossword Scanning App*[4], umožňující naskenování křížkovky z fyzické do elektronické podoby a následné interaktivní řešení na displeji mobilního zařízení. Křížovky je možno skenovat a ukládat pro pozdější řešení, bohužel zde probíhá jenom načtení herního plánu, ale již ne zadání, které musí mít uživatel k dispozici, nebo si jej může prohlédnout z fotografie použité pro úvodní sken. Aplikace také zobrazuje obecné informace o aktuální křížovce, jako je čas strávený řešením, velikost a procentuální stav vyřešení.

3.3.3 Sudoku Photo Solver

V práci Vojtěcha Hrbase – *Řešitel sudoku pro Android*[6], je popisována aplikace, která se zabývá řešením hlavolamu Sudoku⁹ z fotografií zadání. V samotné práci autor rozebírá načtení a zpracování obrazu, následné řešení a promítnutí rozšířené reality, tzn. vyplnění prázdných polí v načteném obrazu zadání.

⁹<https://cs.wikipedia.org/wiki/Sudoku>

Kapitola 4

Moje Aplikace

V následující kapitole se snažím popsat, jaký má moje aplikace přínos, čím obohacuje nasycený trh s aplikacemi pro OS Android a zároveň popisují průchod aplikací a rozebírám obsah jednotlivých obrazovek a dialogových oken.

4.1 Popis

Po objasnění cíle hry Krycí jména a po průzkumu trhu s aplikacemi zabývajícími se tematikou hry Krycí jména je jasné, že se zde objevuje jistá mezera, konkrétně aplikace které opravdu pomáhají přiblížení se vítězství ve hře. Aplikace cílí na hráče doslova prahnoucí po vítězství. Hlavní úkol mé aplikace je tedy dát týmu, vlastníci mobilní zařízení s OS Android výhodu nad týmem protivníků, případně nahradit druhého hlavního agenta chytrým zařízením. Nejlepší možný scénář který může nastat je případ, kdy vlastní mobilní aplikaci jak hlavní agent, tak operativci, stává se tým neporazitelným. Jedinečným přínosem aplikace, oproti použití jen jako tenký klient, napojený na hlavní server je, že vlastní výpočty a vše s hraním spojené probíhá přímo v zařízení samotném, a proto se může stát permanentním partnerem pro hru Krycí jména

4.2 Průběh hry

Po spuštění aplikace se uživatel nachází v hlavním menu. Zde může upravit nastavení, zobrazit přehled aktuálně sdílených her pro možnost načtení, nebo začít hrát ať už z pohledu hlavního agenta nebo operativce.

4.2.1 Nastavení

Tlačítko **SETTINGS**. Po klepnutí na toto tlačítko se zobrazí dialogové okno (obrázek 4.1), ve kterém si hráč může změnit přezdívku, zvolit úroveň nápověd a nakonfigurovat adresy podpůrných serverů třetích stran. Základní přezdívka má tvar `userXXXX`, kde `XXXX` značí 4-ciferné číslo. Pod prvkem pro přezdívku se nachází 3 přepínače. První z nich určuje, zda aplikace bude využívat pro možnosti nápovědy servery, nebo zda bude používat vlastní databázi nápověd. Adresu těchto serverů je poté nutno specifikovat v tabulce vpravo. Druhý a třetí zapínají pokročilou funkcionalitu při výběru nápověd pro operativce, nebo při hádání slov závisejících na nápovědě od hlavního agenta.

Settings

User: user6421 Servers: 10.0.2.2:5000

Use servers ☐ Share: 10.0.2.2:5000

Advanced hints ☒ Hints: athena3.fit.vutbr.cz:8086

Advanced guesses ☐ Guess: athena1.fit.vutbr.cz:8084

Obrázek 4.1: Dialogové okno nastavení.

12:56

Jakubova hra	SPY MASTER	PLAYER
petra	SPY MASTER	PLAYER
user6421	SPY MASTER	PLAYER

Obrázek 4.2: Obrazovka pro načtení hry po spuštění aplikace.

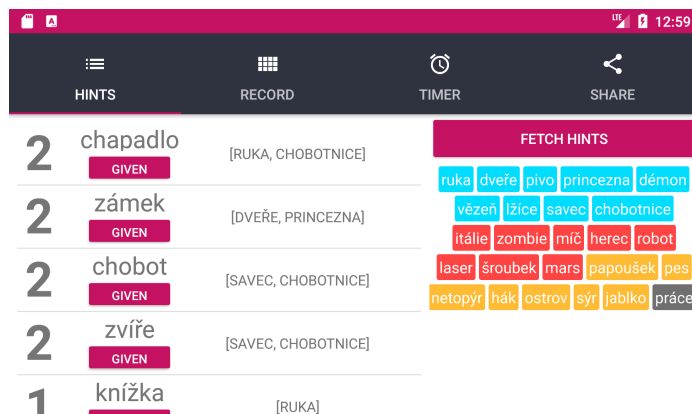
4.2.2 Načtení hry

Tlačítko **LOAD GAME**. Po klepnutí na tlačítko pro načtení hry se otevře nová obrazovka (obrázek 4.2) se všemi aktuálně sdílenými hrami. Každá sdílená hra má časový limit 10 minut, což je doba, po kterou se jeví na serveru jako aktivní a uživatelé si ji mohou stáhnout. Pokud je potřeba hru sdílet i po uplynutí této doby, je potřeba ji nasdílet znovu. Pro stažení a spuštění sdílené hry je potřeba kliknout na jedno ze dvou tlačítek, buď **SPY MASTER** a nebo **PLAYER**, pro zvolení pohledu z jakého bude uživatel hru hrát. Během hraní hry nelze libovolně přepínat pohledy, jediný způsob je právě přes načtení sdílené hry.

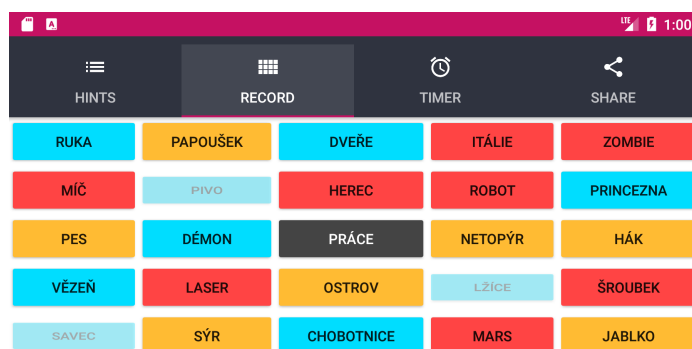
4.2.3 Hra z pohledu hlavního agenta

Tlačítko **SPY GAME** slouží pro spuštění hry z pohledu hlavního agenta. Ihned po stisknutí se zobrazí nová obrazovka vybízející hráče k oskenování herního plánu, poté se přejde na obrazovku k oskenování hrací plochy pro potřeby načtení herních karet. Poslední přípravná obrazovka slouží k ověření správnosti a případnou korekci načtených dat. Pokud se uživatel vyskytuje v nepříznivých světelných podmínkách a skenování ať už herního plánu nebo hrací plochy se nedaří, může tyto dvě obrazovky přeskočit a všechny potřebné údaje vyplnit ručně. Poté se zobrazí nová obrazovka, skládající se ze 4 záložek:

- **Hints** - Obrazovka pro získávání nápověd buď z lokální databáze nebo z pomocných serverů. V pravé části se nachází tlačítko „Fetch Hints“, které provede načtení a



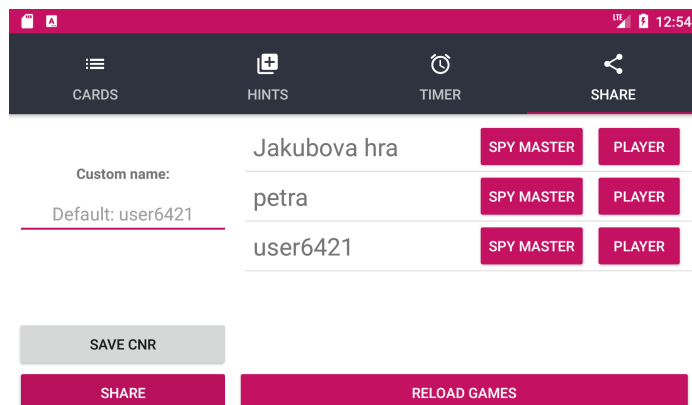
Obrázek 4.3: Obrazovka se záložkou pro zobrazení nabízených nápověd.



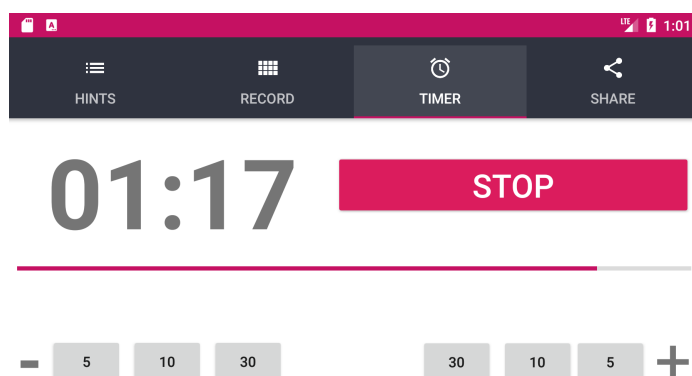
Obrázek 4.4: Obrazovka se záložkou pro záznam všech tahů ve hře. Karty PIVO, SAVEC a LŽÍCE momentálně pulzují, protože byly označeny jako zamýšlené.

zobrazení nápověd. Pod tlačítkem je seznam karet z hrací plochy, podbarvených podle příslušnosti k týmům. (obrázek 4.3)

- **Record** - Obrazovka obsahující 25 tlačítek reprezentujících všech 25 karet položených na hrací ploše. Dlouhým podržením tlačítka se karta označí jako Intended, tedy zamýšlená. Tahem po displeji zdola nahoru se otevře dialogové okno pro zadání vlastní nápovědy operativecům. Po zadání nápovědy je potřeba označit karty, které na základě nápovědy operativeci zvolili. (obrázek 4.4)
- **Timer** - Obrazovka substituující přesýpací hodiny obsažené v herním balení. Umožňují navýšit nebo snížit časový interval a tím udělat hru pomalejší nebo svižnější, ale náročnější.
- **Share** - Obrazovka umožňující sdílení a načtení nové hry a také uložení generovaného CNR záznamu. Záznam lze poté najít v úložišti v nově vytvořené složce *CodeName-Records*. V pravé části lze zvolit vlastní jméno pro každou sdílenou hru. V levé části se nachází seznam her, totožný s obrazovkou Load Game z hlavního menu. Hráč taktéž musí zvolit, pro jaký pohled chce sdílenou hru načíst.



Obrázek 4.5: Obrazovka se záložkou pro sdílení a načítání aktivních her.



Obrázek 4.6: Obrazovka se záložkou pro nastavitelný odpočet času.

4.2.4 Hra z pohledu operativce

Tlačítko **PLAYER GAME** slouží pro spuštění hry z pohledu operativce. Po spuštění módu pro operativce se otevře pouze obrazovka pro naskenování hrací plochy a poté přejde do přípravné obrazovky. Po ověření a korekci, případně při špatných světelných podmínkách celkového zadání všech názvů karet se uživatel přesune do hlavní obrazovky pohledu operativce, ta se podobně jako obrazovka při pohledu hlavního agenta skládá ze 4 záložek:

- **Cards** - Obrazovka pro označování typů karet, po načtení nápověd se zde zvyšuje šance u karty v podobě progressbaru. Přiřazení typu(barvy) kartě proběhne po kliknutí na jedno ze čtyř tlačítek napravo. **A** pro nájemného zabijáka, **B** pro náhodného kolemjdoucího, **BLUE** pro modrý tým a **RED** pro tým červený. Pokud si tým není stoprocentně jistý, může klepnout na šedý kruh v levé části, čímž označí kartu jako zamýšlenou, a může jí případně hádat v dalším kole.
- **Hints** - Obrazovka pro přidávání nápověd získaných od hlavního agenta. Pro přidání je potřeba kliknout na plovoucí tlačítko +1 v pravém dolním rohu. Po přidání nápovědy se provede akce upravující hodnotu karty na záložce Cards. Pokud patří přidaná nápověda mému týmu, zvýší se hodnota všech karet spojených s touto nápovědou o 1. Pokud před přidáním již měly některé karty přiřazenou hodnotu, zmenší se jejich hodnota na 75%. Pokud patří přidaná nápověda soupeři, a je zapnutý pokročilý režim, sníží se u karet souvisejících s nápovědou hodnota o 0,5.
- **Timer** - Stejná obrazovka jako v pohledu hlavního agenta.
- **Share** - Stejná obrazovka jako v pohledu hlavního agenta.

4.2.5 Obrazovka pro načítání herního plánu

Jedná se o aktivitu tvořenou celoobrazovkovým náhledem fotoaparátu. Po úspěšné detekci se na obrazovce vykreslí čtverce odpovídající barvě políčka karty. Pro potvrzení načtení nebo pro přeskočení je potřeba zmáčknout tlačítko v pravém dolním rohu.

4.2.6 Obrazovka pro načítání hrací plochy

Jedná se o aktivitu tvořenou podobně jako při načítání herního plánu jedním velkým celoobrazovkovým náhledem s výpisem již načtených karet a tlačítkem pro pokračování.

4.2.7 Obrazovka pro korekci modelu hry

Na této obrazovce vidí uživatel 25 tlačítek korespondujících s pětadvaceti hracími kartami. Pokud načtení hrací plochy a herního plánu proběhlo bezchybně, uživatel by měl vidět všechna tlačítka obarvená a s popiskem. Pokud se na tuto obrazovku dostal při spuštění hry z pohledu operativce, neproběhlo tedy skenování herního plánu a tlačítka jsou neobarvená. Ve spodní části se nachází ukazatele aktuálního počtu různých typů karet.

4.3 Pomoc

Aplikace poskytuje vícero úrovní pomoci podle nastavení parametrů v dialogovém okně před začátkem hry. Po započetí hry již nastavení této úrovně nejde změnit, aby při domluvě

o použití lehké pomoci hráč nepřepnul potají na vyšší úroveň. Pomoc s nápovědou v základu obsahuje asi 1230 unikátních nápověd, celkem 5730 kusů, s průměrným počtem 14ti nápověd na kartu. Při hraní a záznamu pomocí aplikace Better Codenames se databáze neustále rozrůstá a určitým propojením mezi kartou a nápovědou se zvyšuje hodnota značící kolikrát se toto spojení vyskytlo. Dvojice s nejvyšší hodnotou jsou v následném výběru nápověd upřednostňovány.

4.3.1 Barevný přehled

Nejnižší úroveň, která je přítomná neustále, ovšem pouze v pohledu hlavního agenta, je zobrazení karet v barvách týmu po skupinkách, ale také obarvení karet v náhledu 5x5 pod záložkou **Record**. Hlavní agent tak nemusí neustále střídat pohled mezi herním plánem a hrací plochou, orientovat se ve sloupečcích a řádcích a hledat karty náležící právě jeho týmu. Podobný soupis, ale již seskupený podle barev má i v záložce **Hints** napravo.

4.3.2 Nápovědy

Dalším způsobem pomoci hráči je nabízení slov určených jako nápověda operativcům, na základě kterých se následně snaží uhádnout všechny karty týmu. Protože hlavní agenti mají přehled o všech kartách ve hře, lze v nastavení zvolit optimální metodu nabízení slov. Vybrané nápovědy se poté seřadí podle počtu karet, které se dají pomocí nich uhádnout. Na výběr jsou tyto metody a jejich kombinace:

- Výběr všech možných nápověd z databáze, které souvisejí s danou kartou
- Výběr všech možných nápověd pro kartu s tím rozdílem, že nápovědy, které sdílí i soupeřovy karty jsou skryté
- Výběr všech možných nápověd bez nápověd, které jsou schopny ovlivnit výběr na kartu nájemného zabijáka
- Výběr všech možných nápověd pro kartu bez nápověd společných s kartami náhodných kolemjdoucích

Aplikace podporuje i napojení na cizí server, který vybírá možné nápovědy pomocí shlukové analýzy, této části jsem se ovšem sám nevěnoval a proto ji dále nebudu rozvádět. Použití tohoto serveru je možné aktivovat v dialogovém okně v hlavním menu po kliknutí na ikonu nastavení.

4.3.3 Hádání karet

Třetí způsob, jak uživateli ulehčit hru je zamýšlen do pohledu operativce. Operativec postupně zadává všechny nápovědy, které hlavní agenti vypustí a značí u nich příslušnost k týmu. Po přidání nápovědy jeho týmu se v případě shody se záznamy v databázi zvýší v záložce **Cards** hodnota karty o 1. Následně se karty podle této hodnoty seřadí, takže tento proces ulehčí uživateli výběr vhodné karty, samozřejmě se ale může rozhodnout podle sebe a vybrat kartu jinou. Během střídání tahů a přidávání nových nápověd dochází k procesu degradace nápověd. To znamená že dříve přidané nápovědy ztrácí svoji váhu, aby bylo jednoduché rozlišit karty související s aktuální nápovědou. Při povolení pokročilejšího módu se po přidání nápovědy pro soupeřův tým provádí degradace karet, které jsou podle záznamu v databázi ve spojení právě s touto protivníkovou nápovědou. Stejně tak, jako při výběru ideální nápovědy je možné využít server počítající hodnotu jednotlivých karet.

4.4 .CNR záznam

S každou započatou hrou se vytvoří struktura odpovídající formátu interně nazvaným *Code Names Record*. Strukturu lze vyexportovat do souboru a následně může posloužit k analýze tahů, učení a zlepšování umělé inteligence například při tvorbě robotického spoluhráče ke hře Krycí jména.

Hlavička formátu CNR se skládá z několika položek:

- **T (title)** - název záznamu jedné hry, obsahuje uživatelské jméno a časovou známku identifikující začátek hry
- **1** - 9 slov začínajícího týmu
- **2** - 8 slov nezačínajícího týmu
- **A (assassin)** - slovo nájemného zabijáka
- **B (bystanders)** - 7 slov náhodných kolemjdoucích

Pomyslné tělo formátu CNR zaznamenává každý tah obou týmů zvlášť. Položky, které může obsahovat jsou:

- **C (clue)** - nápověda hlavního agenta týmu, který je momentálně na tahu, skládá se ze slova a čísla
- **I (intended)** - slova zamýšlená hlavním agentem
- **G (guesses)** - seznam hádaných slov v aktuálním tahu, pokud tah končí označením náhodného kolemjdoucího nebo nájemného zabijáka, přidá se na konec atribut (A) nebo (B)
- **D (discussion)** - slova, o kterých si tým myslí, že mají spojitost s obdrženou nápovědou
- **R (rule out)** - speciální případ diskuse o slovech vyloučených nápovědou s číslovkou 0, speciální zvýraznění v záznamu jako rule out momentálně není podporováno
- **E (exclude)** - doporučení pro vyloučení dané nápovědy, například pokud je nápověda v rozporu s pravidly hry
- **N (note)** - poznámka
- **M (multi-word expression)** - kontext, který má ukázat souvislost mezi slovy, zadání kontextu bohužel není momentálně podporováno
- **L (layout)** - popis momentálního stavu hry ve tvaru:
 - slova hrající týmu
 - slova náhodných kolemjdoucích
 - slova týmu soupeře
 - slovo nájemného zabijáka

Kapitola 5

Implementace

Tato kapitola popisuje způsoby a technologie, které jsem použil nebo uvažoval o použití při vývoji aplikace *Better Codenames*. Přestože jsem dlouho váhal zda použít jazyk Java nebo webové technologie pro jednoduché rozšíření aplikace z mobilních zařízení i na stolní počítače a ostatní platformy, rozhodl jsem se pro Javu z důvodu podpory knihoven pro práci s obrazem a s menším důrazem na serverový back-end. K vývoji jsem použil oficiální vývojové prostředí pro Android – *Android Studio 3.1* zobrazené na obrázku 5.1. Při tvorbě projektu je potřeba vybrat minimální podporovanou verzi SDK, na základě čeho je zvolena minimální verze operačního systému, která je potřeba pro bezproblémový chod aplikace. Rozhodl jsem se pro API s číslem 25, toto číslo odpovídá Androidu ve verzi *5.0 Lollipop*. Podle oficiálních statistik¹ běží na verzi 5.0 nebo vyšší okolo 85% zařízení s OS Android, což se mi zdá dostačující, o řadu nižší Android 4.x zabírá nyní jen s nadsázkou desetinu trhu. Vzhledem k stáří těchto verzí jsem nemožnost spuštění na těchto zařízeních schopen akceptovat.

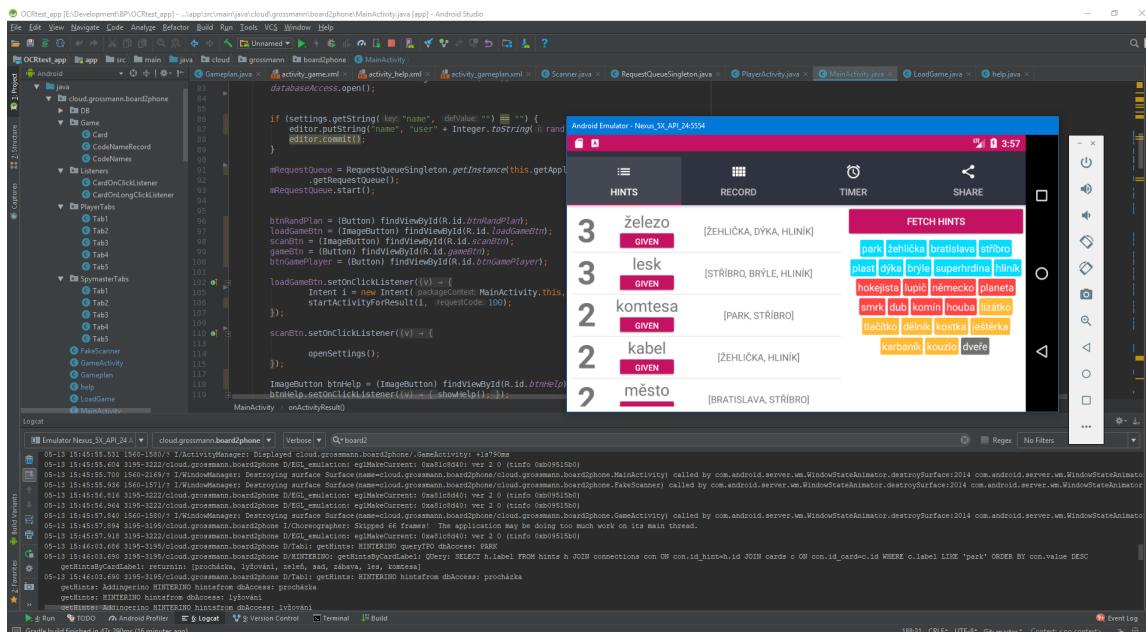
5.1 Vývoj pro OS Android

Základem pro vývoj Android aplikací je Android SDK, neboli Software Development Kit. Tento nástroj umožňuje uživateli překládat aplikaci do požadovaného formátu jen za pomoci příkazové řádky. Protože ale Android Studio obsahuje SDK manažer, používal jsem při vývoji všechny výhody grafického uživatelského rozhraní jako jednoduché aktualizování verzí, tvorbu .APK souborů, debugování pomocí nástroje logcat, ale s poslední verzí i přehledného a intuitivního průzkumníka souborů v připojeném, popřípadě emulovaném zařízení. Pro potřeby podpory OpenCV jsem musel do projektu zařadit i Android NDK, neboli Native Development Kit, který umožňuje implementovat aplikaci v nativním kódu, pomocí jazyků jako jsou C a C++.

5.1.1 Java

Dlouhou dobu byla Java jediná možná cesta pro vývoj nativních Androidích aplikací. Java je vyvíjena firmou Oracle (dříve Sun Microsystems), jedná se o objektově orientovaný programovací jazyk. Pro spuštění se používá překlad do Java byte-kódu (většinou s příponou .jar nebo .class) a ten se následně spouští na Java Virtual Machine který jej interpretuje. Tímto způsobem je zajištěna jednak multiplatformnost při které se kód chová na různém

¹<https://developer.android.com/about/dashboards/>



Obrázek 5.1: Vývojové prostředí Android Studio 3.1 se spuštěným emulátorem a debugovacím nástrojem Logcat.

hardware stejně, ale také možnost spouštět kód s rozdílnou syntaxí než Javovskou, pokud je výsledný byte-kód validní. Pro účely použití Javy v systému Android byla vytvořena nová knihovná třída, namísto stávající Java Class Library. Na zařízeních s OS Android se místo JVM využívá Dalvik Virtual Machine, který místo klasických `.jar` požaduje soubory `.dex`, tedy Dalvik Executables. Pro detailní popis jak funguje vývoj aplikací pro Android je popsán v oficiální dokumentaci od společnosti Google^[3].

5.1.2 Kotlin

Kotlin² je staticky typovaný programovací jazyk pro tvorbu moderních multiplatformních aplikací představený roku 2011. Hlavní výhodou je možnost překladač do byte kódu běžícího nad klasickým JVM³, ale také „překlad“ do JavaScriptu. Syntaxí se Kotlin podobá Javě a Pythonu. Google zařadil Kotlin mezi oficiální programovací jazyky pro vývoj Android aplikací v květnu 2017⁴.

5.1.3 .NET

Pro vývoj Android aplikací lze nečekaně použít i technologii .NET od Microsoftu. Pomocí technologie Xamarin.Android lze dosáhnout výkonu srovnatelného s nativními aplikacemi, ale zde jsou aplikace psané pomocí jazyku C# a F#. Hlavní vývojové prostředí je nyní nově integrovaný Xamarin do Microsoftiho Visual Studia.

²<https://kotlinlang.org/>

³Java Virtual Machine

⁴<https://www.theverge.com/2017/5/17/15654988/google-jet-brains-kotlin-programming-language-android-development-io-2017>

5.1.4 Webové technologie

Aplikace pro Android lze vytvářet také pomocí webových technologií HTML, CSS a JavaScriptu. Výhodou těchto aplikací je snadná přenesitelnost mezi platformami, tedy současný vývoj i pro zařízení s iOS, Blackberry10, již dosluhující Windows Phone8.1, Linux a OS X. Výkonově se ovšem stále nedají srovnat s nativními aplikacemi. Na trhu se vyskytuje hlavní a populární open-source framework pro mobilní vývoj pomocí webových technologií, a to Apache Cordova postavený na Node.js. Obsahuje nespočet rozšíření pro usnadnění vývoje umožňující vytvořit funkční aplikaci v řádech minut. Například Adobe PhoneGap⁵ nebo Ionic⁶ postavený na AngularJS.

5.2 Základní kameny Java Android aplikací

Pro vývoj Android aplikací a následné sestavení funkční aplikace je potřeba respektovat určité zásady. Každá aplikace se skládá z **aktivit, layoutů, assetů, zdrojů a samotných tříd**. Užitečné informace podobné zde popsaným a mnohem více v knihách *Vývoj aplikací pro Android*[8] a *Programming Android*[11].

5.2.1 AndroidManifest

Manifest obsahuje seznam všech aktivit a oprávnění, které aplikace potřebuje k chodu. V mém případě povolení používat kameru, číst a zapisovat do externího úložiště a oprávnění používat internet. Nachází se zde i název aplikace, ikona, nastavení grafických prvků, například jestli zobrazovat hlavní lištu atd. Jedna z mnoha aktivit je zde uvedena jako hlavní, tedy aktivita která se vykreslí při spuštění aplikace.

5.2.2 Aktivita

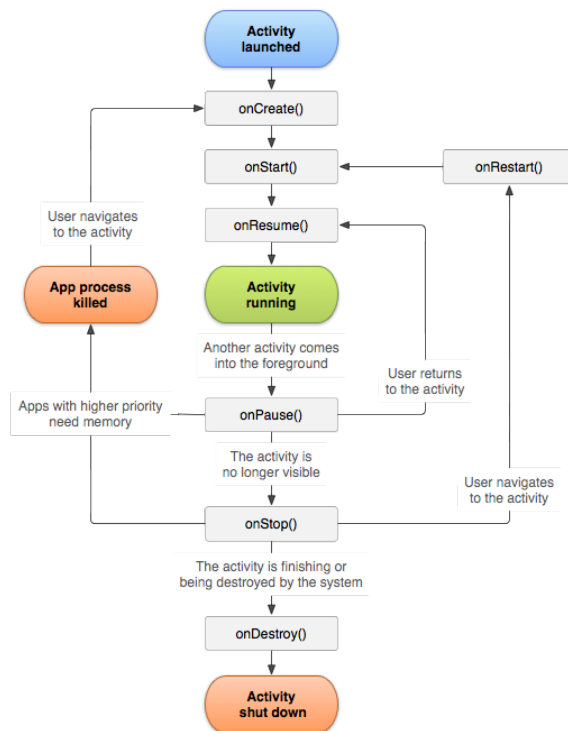
Každé obrazovce, kterou chce programátor na zařízení zobrazit odpovídá jedna aktivita. Aktivita dědí ze třídy **Activity** a potřebuje přepsat metodu **onCreate()**, která zastává funkci pomyslného konstruktoru obrazovky. Aby bylo na obrazovce co zobrazit, je aktivitě přiřazen Layout. Aktivita mají přesně definovaný životní cyklus, tedy dobu po kterou existují, během které přechází aktivita mezi různými stavy. Životní cyklus aktivity začíná v momentu zavolání funkce **onCreate()** a končí v okamžiku zavolání metody **onDestroy()**. S tímto přichází i mnohé nevýhody, které jsou ovšem dobře opodstatnělé, jedna z nich například je, že při změně orientace obrazovky se aktivita zastaví a spustí znovu. Tím sice přijdeme o všechna data, která byla v aktivitě vytvořena od jejího vzniku, na druhou stranu ale můžeme použít rozdílný layout vytvořený exkluzivně pro orientaci na šířku.

5.2.3 Fragment

Fragment reprezentuje část uživatelského rozhraní, dovoluje programátorovi kombinovat větší počet fragmentů do jedné aktivity a tím vytvořit dynamické prostředí bez nutnosti vytvářet další aktivity. Hlavním důvodem vývoje fragmentů byl růst využívání OS Android i na tabletech, kde se ukázaly možnosti obyčejných aktivit jako nedostatečné. Fragment musí být vždy obsažen v nějaké aktivitě a jeho životní cyklus je maximálně tak dlouhý, jako životní cyklus hostitelské aktivity, přičemž lze fragmenty spouštět a zastavovat dle

⁵<https://phonegap.com>

⁶<https://ionicframework.com>



Obrázek 5.2: Životní cyklus Aktivity

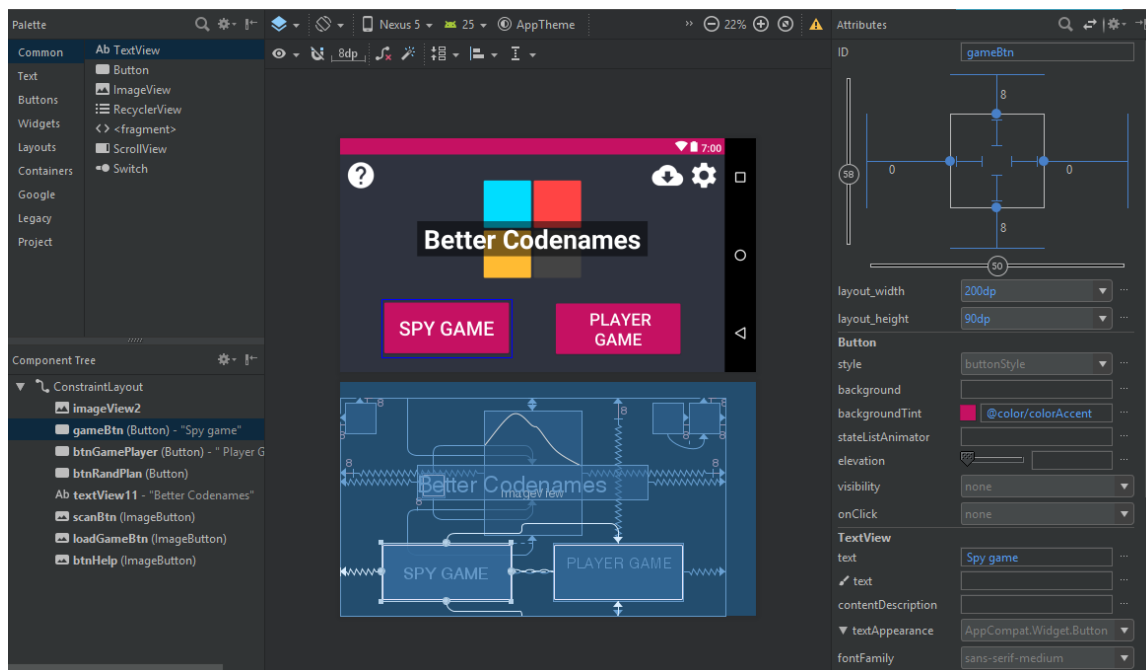
libosti. V mé aplikaci používám fragmenty pro potřeby záložek v uživatelském rozhraní, kdy je každá záložka reprezentována vlastním fragmentem.

5.2.4 Layout

K návrhu grafického rozhraní se v Androidu používá značkovací jazyk XML. Každá obrazovka musí obsahovat jeden hlavní layout, ve kterém se nacházejí další grafické prvky jako tlačítka, textová pole a jiné elementy. Nejpoužívanější typy hlavních layoutů jsou například

- **LinearLayout**, který skládá jednotlivé elementy vedle sebe nebo pod sebe, v závislosti zda se jedná o vertikální nebo horizontální layout
- **TableLayout**, který umožňuje z elementů seskládat tabulku, stejného efektu lze dosáhnout i kombinací několika LinearLayoutů
- **GridLayout**, který je novější náhrada za TableLayout dokáže totožné věci jako dříve zmíněný, ale opravuje problémy s kompatibilitou a přináší intuitivnější rozvržení prvků
- **ConstraintLayout**, který umožňuje programátorovi jednoduše a přesto efektivně vytvářet responzivní uživatelské rozhraní pomocí tzv. propojení mezi elementy. Poté lze nastavit zda se má měnit velikost nebo pozice prvku na obrazovce. Výhodou tohoto layoutu je, že elementy jsou závislé vůči sobě a ne vůči pomyslné kostře obrazovky

Pro přehled uvedu i krátký seznam elementů, které lze při tvorbě uživatelského rozhraní využít a které jsem již také sám použil. Jedná se například o *Button*, *ImageView*,



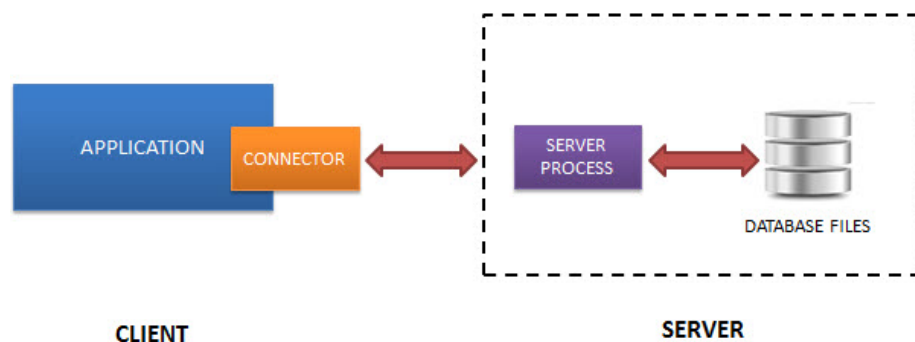
Obrázek 5.3: Grafický návrh uživatelského prostředí pomocí ConstraintLayout v editoru Android Studio.

TextView, *EditText*, *CheckBox*, *RadioButton*, *Switch*, *ProgressBar*, *RatingBar*, *MapView* a také elementy přidané do aplikace pro podporu zpracování a vykreslování obrazu *CameraGLSurfaceView* a *JavaCameraView*.

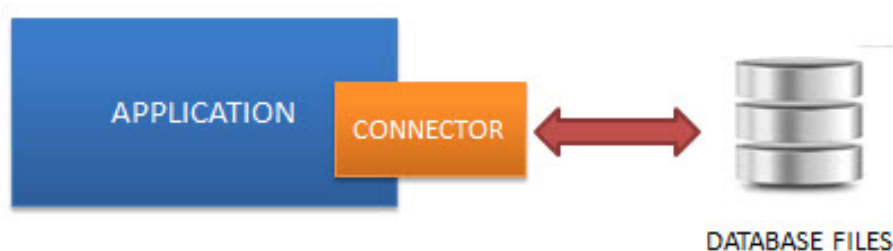
5.3 Uchování a získání nápověd

Všechna data spojená s výběrem nápověd musí být někde uchována. Momentální slovní zásoba není největší, ale s aktivním používáním aplikace je možné, že počet nápověd se zněkolikanásobí. Z tohoto důvodu se mi zdálo nežádoucí a také náročné na zdroje tyto data neustále držet v paměti jako kolekci. Také by se za pomoci serializace muselo vyřešit uložení při vypnutí a následné načtení při zapnutí aplikace. Jako vhodné řešení se tedy nabízí uložení dat do databáze, která by se skládala ze tří tabulek. V první je přesně 400 záznamů, každý odpovídá jedné hrací kartě a nese informaci o jejím nápisu. Druhá tabulka obsahuje všechny možné nápovědy, a každá je zde právě jedenkrát. Poslední tabulka slouží pro uložení spojení mezi kartou a nápovědou, obstarává vazbu M ku N a jako bonus také číselné ohodnocení tohoto spojení. Čím vyšší číslo, tím více je spojení karty s nápovědou populární a operativci podle něj dokáží identifikovat zamýšlenou kartu.

Hlavním důvodem, proč jsem se rozhodl implementovat nápovědy pomocí pevně zadáných slov, bez využití tzv. „buzzwordů“ jako jsou strojové učení nebo umělá inteligence bylo, že existuje neskutečné množství slov, kterými by šly popsat snad všechny kombinace 200 karet ve hře. V ideálním případě by se nacházelo jedno klíčové slovo související s daným počtem karet, zaručující stoprocentní šanci na uhádnutí zamýšlených předmětů. Vytvoření takovéto databáze a nalezení všech možností by bylo samozřejmě velice pracné. Zde se přímo nabízí k tomu použít stroj zpracovávající text podle určitých pravidel za nás.



Obrázek 5.4: Klasický způsob fungování modelu aplikace a databáze. zdroj <http://www.sqlitetutorial.net/what-is-sqlite/>



Obrázek 5.5: Způsob jaký využívá SQLite, za povšimnutí stojí přístup přímo k databázovým souborům. zdroj <http://www.sqlitetutorial.net/what-is-sqlite/>

Jenže tato silná stránka ultimátních slov se může negativně promítnout do druhé části hry, kde operativci hádají karty. Pokud operativce neví, co si pod daným slovem představit, stávají se z nápořád pouze rozptylující shluky písmen. V aplikaci tak nejen využívám předpřipravenou sadu slov, ale tuto sadu neustále rozšiřuji, vazby mezi nápořádami a slovy se upravují, podle skupiny lidí v jejichž přítomnosti je pomocník používáný. Často používané kombinace se tak průběhem času dostanou do popředí.

5.3.1 SQLite

Po rozhodnutí ukládat data do databáze ovšem nastal problém, kterou databázovou službu použít. Při použití kterékoliv databáze uložené na vzdáleném serveru by nebylo možno hrát bez připojení k internetu a proto se jako skvělý kandidát ukázal relační databázový systém SQLite⁷. Velkou výhodou je absence nutnosti spouštět další proces, který by se staral o přístup k databázi a návrat hodnot, protože SQLite zastává bezserverovou architekturu. Rozdíly mezi klasickým SQL serverem a SQLite jsou demonstrovány na obrázcích 5.4 a 5.5. SQLite dokáže pomocí jednoduchého rozhraní přistupovat k datům uloženým v databázovém souboru a vykonávat nad ním klasické SQL dotazy. Více o SQLite lze najít v oficiální dokumentaci [1].

⁷<https://www.sqlite.org/about.html>

5.3.2 Přístup k datům

Data nyní nejsou uložena v aplikaci, ale v databázovém souboru, proto je potřeba třída `DatabaseAccess` vykonávající SQL dotazy nad samotnou instancí databáze pro manipulaci a získávání dat ze souboru. Tyto dotazy obaluje do metod s klasickými návratovými hodnotami, v mém případě většinou listy řetězců. Pro potřeby mé aplikace používám tři druhy dotazů. Dotazy vyhledávací, dále aktualizovací dotazy pro potřeby inkrementaci ohodnocení spojení mezi kartou a nápovědou, a v poslední řadě dotazy vkládací pro přidání nových záznamů do tabulky.

```
SELECT jmeno_sloupce FROM jmeno_tabulky;  
UPDATE jmeno_tabulky SET hodnota2='10';  
INSERT INTO jmeno_tabulky values('1');
```

V samotné aplikaci se poté zavolá jedna z přístupujících funkcí obalujících tyto SQL dotazy a zpracovávající vrácené hodnoty. Příklad takovéto funkce je následující:

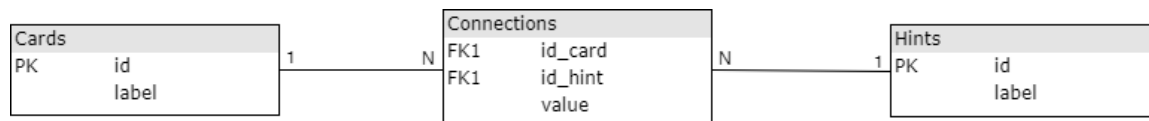
```
public List<String> getHints() {  
    List<String> list = new ArrayList<>();  
    Cursor cursor = database.rawQuery("SELECT label FROM hints", null);  
    cursor.moveToFirst();  
    while (!cursor.isAfterLast()) {  
        list.add(cursor.getString(0));  
        cursor.moveToNext();  
    }  
    cursor.close();  
    return list;  
}
```

Používat databázi jen pro jednoduché uložení dat je ovšem neefektivní, pomocí pokročilých dotazů jsem schopený po spojení více tabulek získat pouze záznamy s požadovanými vlastnostmi. Tímto odpadá nutnost nahrát si všechny záznamy do paměti a provádět filtraci až nad nimi. Mezi pokročilé dotazy patří spojení tabulek. Příkaz pro spojení dvou tabulek se jmenuje `JOIN` a mezi podporované typy v SQLite patří `INNER JOIN`, také nazývaný jako jednoduché spojení, který provede průnik dvou tabulek. Dalším typem je `LEFT OUTER JOIN`, který vrátí všechna data z tabulky 1 spojené s daty z tabulky 2, která jsou společná pro obě tabulky. SQLite nepodporuje `RIGHT OUTER JOIN`, protože zaměněním pořadí tabulek lze docílit stejného výsledku s levým vnějším spojením. Poslední podporovaný typ je `CROSS JOIN`, který spojí každý řádek tabulky 1 s každým řádkem tabulky 2, výsledkem tedy bude obrovské množství dat, pokud má tabulka 1 záznamů 5 a tabulka 2 záznamů 10, výstupem bude tabulka o 50 záznamech. Pro potřeby mé aplikace používám pouze jednoduché spojení.

Po spojení více tabulek je potřeba z velkého množství dat dostat pouze záznamy, které odpovídají určité podmínce, v mém případě například souvislost s nápovědou. K tomuto účelu poslouží příkaz `WHERE` podmínka. Pokud je potřeba, lze skládat různé podmínky za sebe pomocí operátorů `AND` a `OR`, kdy první z nich vrátí záznam při splnění obou podmínek a druhý v případě pokud je alespoň jedna z podmínek vyhodocena jako pravdivá. Příkaz `WHERE` tak zastává velice důležitou funkci a jeho správným umístěním v určitých případech v SQL dotazu můžeme dosáhnout i jisté optimalizace. Pokud se příkaz `WHERE` provede již na první tabulce před jejím spojením s druhou, nemusí databáze vytvářet kompletní tabulku

a poté filtrovat všechny záznamy, ale po spojení již vznikne výsledná tabulka, přičemž filtrování se provedlo jen na záznamech první tabulky.

Za zmínku také stojí příkaz `ORDER BY`, který umožňuje řadit záznamy výsledku vzestupně nebo sestupně a příkaz `LIMIT`, který omezí počet vrácených záznamů.



Obrázek 5.6: Diagram návrhu tabulek databáze.

5.4 Rozpoznávání

V knize Počítačové vidění[12] se uvádí, že "Počítačové vidění je disciplína, která se snaží technickými prostředky alespoň částečně napodobit lidské vidění.". Toho se snažím dosáhnout i ve své práci při rozpoznání herních plánů, kdy se snažím docílit toho, aby aplikace měla přehled jaké barvy se v plánu vyskytují. V neposlední řadě, aby odhalila všechna slova nacházející se na kartách hrací plochy.

5.4.1 Knihovny pro práci s obrazem

OpenCV neboli Open Computer Vision, původně vyvinutá společností Intel je sada open-source knihoven používaných pro počítačové vidění. Prvotní verze 1.0 byla do světa vypuštěna již roku 2006, verze 2.0 pak o tři roky později. Tato verze přidala ke stávajícím funkcím mnoho nového, upravila rozhraní vytvořené v C++ a zvýšila výkon jednotlivých operací, převážně na vícejádrových systémech. Mezi funkce, pro které lze OpenCV využít se řadí například rozpoznávání tváří a gest, interakce člověka s počítačem, prostorové vidění počítače za pomoci dvou a více kamer, záznam pohybu, rozšířená realita a mnohé další.

OpenCV je sice psané v C++, ovšem z důvodu velké popularity je jej možné používat pomocí wrapperů i v jazyce Python a také Java, která je důležitá pro moji aplikaci.

BoofCV⁸ je open-source knihovna kompletně napsaná v jazyce Java, provádějící počítačové vidění v reálném čase. Soustředí se na široké rozmezí funkcí, jako nízkourovňové zpracování obrazu, vytváření 3d obrazu z více 2d fotografií, kalibraci kamery, rozpoznávání textu a detekce či sledování objektů. BoofCV obsahuje velice rychlou implementaci SURF⁹, tedy funkce obstarávající rozpoznání objektů, klasifikaci a 3d rekonstrukci. Z tohoto důvodu jsem uvažoval o BoofCV jako o vhodném kandidátovi pro provádění *template matchingu* při rozpoznávání herních plánů.

Google Mobile Vision API¹⁰ poskytuje framework pro nacházení objektů ve fotkách a videu. Framework obsahuje detektory, které dokáží nalézt pozici a vizuálně popsat objekty v obrázcích nebo snímcích videa, a také API pro získávání pozice těchto objektů ve snímku. API momentálně zahrnuje rozpoznávání obličejů, čárového kódu a textu. Hlavní výhodou je, že API je zcela zdarma a volně k použití, a to i bez nutnosti internetového připojení.

Google Cloud Vision API je placená online služba¹¹, která schovává nejružnější funkce a pokročilé strojové učení za jednoduché RESTové API, zpracovávající obrázky za-

⁸<http://boofcv.org>

⁹https://en.wikipedia.org/wiki/Speeded_up_robust_features

¹⁰<https://developers.google.com/vision>

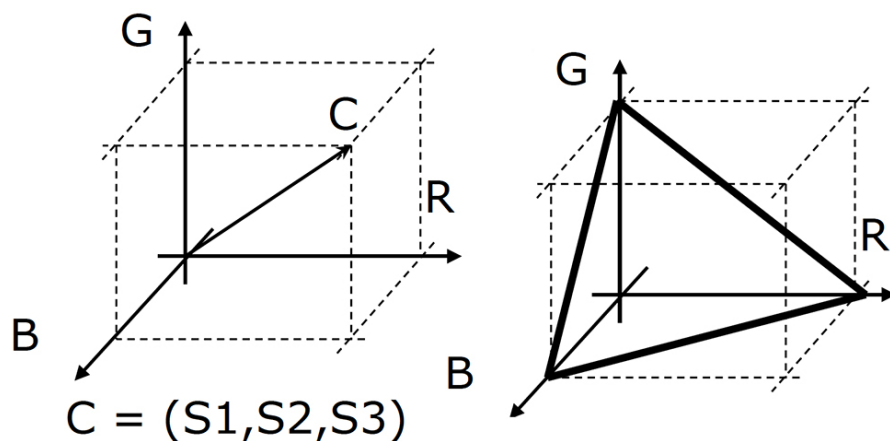
¹¹<https://cloud.google.com/vision>

slané v obsahu požadavku nebo přímo z cloudového úložiště Google Disk. Služba dokáže klasifikovat obrázek do tisíců kategorií, nebo detekovat individuální objekty. Detekce obličejů a automatická detekce textů včetně rozpoznání jazyka je samozřejmostí. Dále dokáže služba určit, zda obrázek obsahuje materiál určený pro dospělé, najít loga známých firem a nebo využít sílu internetu a nalézt podobné obrázky. Bohužel je tato služba zdarma pouze do 1000 provedených úkonů za měsíc, přičemž pokud na jeden obrázek použijí detekci obličejů a zároveň rozpoznání textu, budou mi započítány 2 úkony.

Tesseract OCR Engine pro optické rozpoznávání znaků, který bych zajisté mohl využít při tvorbě pro potřeby načítání hrací plochy. Výhodou je podpora UTF-8 a rozeznání více než 130 jazyků, mezi které patří i čeština. Použitím bych ovšem posunul práci více na úroveň samotného rozpoznávání a ne na podporu při hraní. Náročnost a příklad použití lze vidět zpracovaný v diplomové práci *Zpracování obrazu v zařízení Android - detekce a rozpoznání vizitky* od Martina Krčmáře[7].

5.4.2 Barevné modely

Každý bod obrazu má určitou barvu, v OpenCV reprezentovanou vektorem, jehož hodnoty odpovídají reprezentaci barvy podle zvoleného barevného modelu.

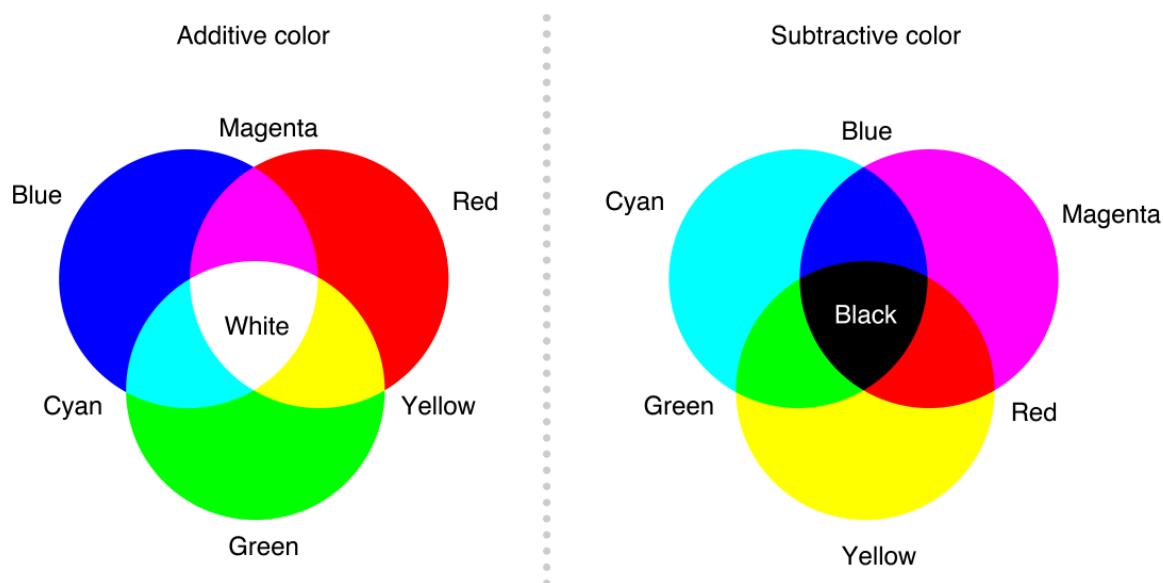


Obrázek 5.7: Zobrazení reprezentace barvy v jednotkové krychli pomocí vektoru. zdroj <http://obsessive-coffee-disorder.com/tag/hsv>

Barevné modely se dělí na aditivní a subtraktivní. V aditivním modelu se jednotlivé složky barvy sčítají, typickým příkladem aditivního modelu je například model RGB¹². Naopak v subtraktivním modelu barev se od sebe barevné složky odečítají, pro potřeby subtraktivního modelu je nutno použít doplňkové barvy základního aditivního modelu RGB, tedy CMY¹³. Platí, že aditivním smícháním barvy a barvy k ní doplňkové získáme barvu bílou, zatímco subtraktivním smícháním barvy a barvy k ní doplňkové získáme barvu černou. Modely RGB a CMY lze znázornit buď na jednotkové barevné krychli, kde každý z vrcholů v osách souřadného systému znázorňuje jednu z barev RGB, v protilehlých vrcholech jsou barvy doplňkové, CMY, a ve zbývajících dvou se nachází bílá a černá, popřípadě lze modely vyjádřit pomocí Vennových diagramů.

¹²Red Green Blue

¹³Cyan Magenta Yellow



Obrázek 5.8: Znázornění modelů RGB a CMY pomocí Venových diagramů. zdroj https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/csintro/csintro_colorspace/csintro_colorspace.html

HSV¹⁴ model je člověku nejvíce intuitivní svým popisem barev, skládajícím se z barevného tónu, sytosti barvy a jasů. Barevný tón udává nejvýraznější spektrum barvy, jas příměs bílého světla (v lidském popisu jak moc je barva světlá) a nakonec sytost barvy udává jak moc je barva míšená s jinými z barev. Model HSV je většinou znázorněn pomocí šestibokého jehlanu kde každý z vrcholů na obvodu podstavy odpovídá jedné ze šesti čistých barev na vrcholech jednotkové krychle. Barevný tón může nabývat hodnot 0° až 360° a měří se od osy S proti směru hodinových ručiček. Osa S prochází vrcholem značícím červenou barvu. Pomocí středové osy se značí jas i sytost, na vrcholu jehlanu v hodnotě 0 se nachází černá barva, ve středu podstavy v hodnotě 1 pak barva bílá. Detailní popis modelu HSV je důležitý, z důvodu použití při rozpoznávání barev na herním plánu.

RGB model, jak z názvu vyplývá, tvoří tři základní barvy – červená, modrá a zelená. Jedná se o model aditivní, běžně se používá v zobrazovací technice jako jsou displeje mobilů, monitory a televizory.

CMY model tvořený barvami azurová, purpurová a žlutá je subtraktivní model využívaný hlavně při tisku, konkrétně v tisku ale nevynikne černá úplně dokonale, a proto se k barvám přidává ještě černá. Kompletní označení pro model používaný v tiskařském průmyslu je CMYK.

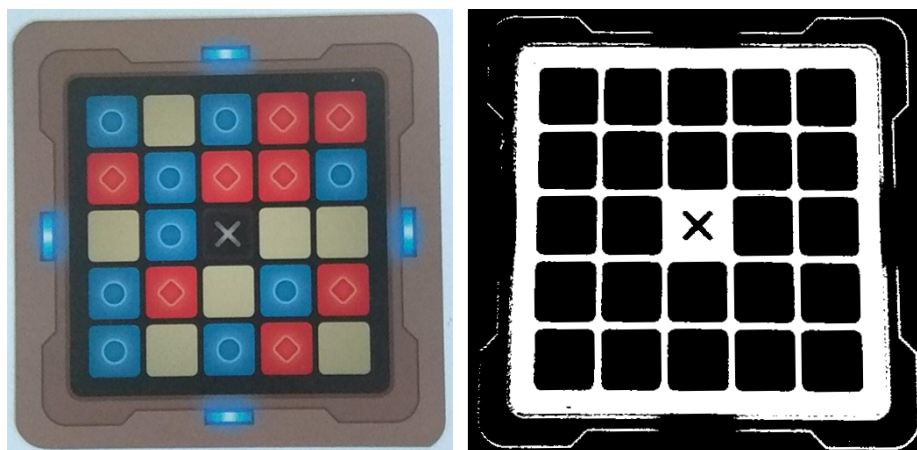
HSL¹⁵ model je velice podobný modelu HSV, ovšem odstraňuje některé jeho nedostatky. Udává se, že chápání sytosti a světlosti jako dvě nezávislé veličiny je pro člověka intuitivnější. Model, znázorněný jako dva rovinně souměrné jehlany HSV dotýkající se podstavami, obsahuje 6 hran čistých barev, barevný tón se měří pomocí úhlu s osou S a středová osa udává světlost. Popis barevných modelů HSV a HSL je přehledně zpracovaný v článku na Obsessive Coffee Disorder[2].

¹⁴Hue Saturation Value

¹⁵Hue Saturation Light

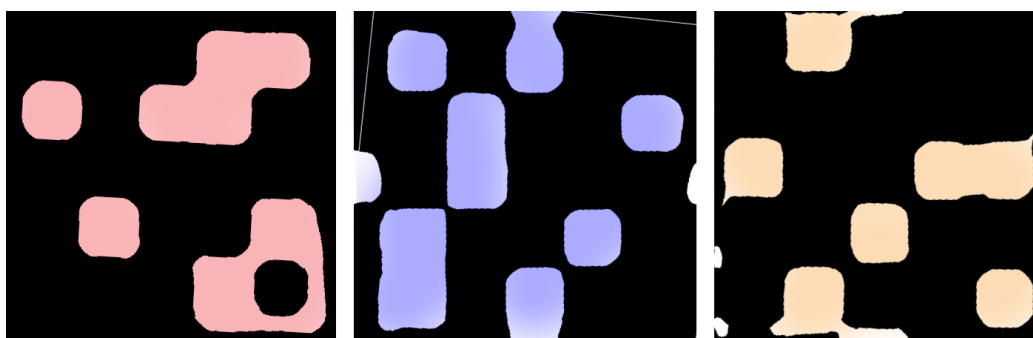
5.4.3 Rozpoznávání herního plánu

Rozpoznávání herního plánu se ve zdrojových souborech nachází ve třídě `Gameplan.java`. Obraz získaný z kamery zařízení uloží do matice, kterou převedu z modelu RGB do dříve zmiňovaného HSV. Obraz rozostřím pro vyhlazení šumu a aplikuji na něj prahování¹⁶. Výstupem je jednobarevná matice z které jde jasně poznat hlavní mříž herního plánu. Tuto mříž jsem schopen rozpoznat a udělat výřez originálního obrazu v modelu HSV ve stejné oblasti.



Obrázek 5.9: Ukázka originálního herního plánu a výstup po prahování s viditelnou mříží.

Detekce jednotlivých políček poté probíhá téměř totožně. Protože se uprostřed většiny políček nachází rušivé obrazce, na výřez aplikuji mediánový filtr¹⁷ a také vlastní funkci, která nastaví hodnotu jasu daného bodu na maximum. Užitečné znalosti, jako je použití filtru před prahováním jsem se dozvěděl z knihy *OpenCV 2 computer vision application programming cookbook*[9]. Díky vlastnostem HSV modelu tak vynikne dominantní barva. Pomocí prahování získám podobné jednobarevné matice pro červený tým, modrý tým i náhodné kolemjdoucí. Toto řešení bohužel nelze použít pro nalezení políček nájemného vraha, protože černá barva je reprezentována pomocí jasu v hodnotě 0.



Obrázek 5.10: Trojice výstupů získaných pro rozeznání políček. pozn. Pro přehlednost jsem dodatečně obarvil bílá místa v grafickém editoru.

¹⁶[https://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing))

¹⁷<https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html#medianblur>

Takto připravené černobílé matice předávám třídě **Plan**, která uskuteční převod do serializovatelného tvaru, pomocí algoritmu, který se podobá níže popsanému algoritmu, používanému při načítání hracích karet.

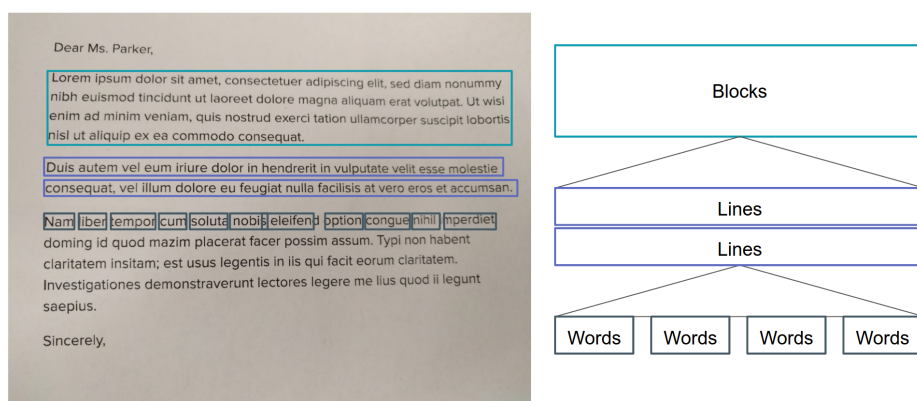
5.4.4 Rozponávání textu

Rozpoznávání textu je proces detekování textu v obraze a následné rozpoznání nelezeného textu. Tvořit vlastní OCR čtečku a obstarávat všechny náležitosti sám mi přišlo v momentě vývoje aplikace nad rámec, ale také bych jen znovu vymýšlel kolo. O použití některé z OCR knihoven jsem měl tedy hned jasno. Pro mé potřeby rozpoznávání textu se nejlépe osvědčilo použití Text Recognition API z Google Mobile Vision, které dokáže rozpoznat text skládající se ze znaků latinky v reálném čase, bez potřeby být připojený k internetu. V momentě kdy je detekován text, rozpoznávač označí bloky textu a rozdělí je na řádky nebo samotná slova. Knihovna bohužel nepodporuje češtinu, a proto se mnohdy na výstupu objevovaly patvary slov, napůl s diakritikou a napůl bez, nebo se dokonce slova na prvních pár pokusů ani nerozpoznaly.

Proto provádím ještě dodatečnou korekci výstupu v podobě hledání shody před i po převodu na slovo bez diakritiky. S každým takto získaným slovem jsem také schopný zjistit jeho pozici na obrazovce. Bohužel málokdy nastane rozpoznání všech 25 karet v jednom snímku. Proto z rozpoznávaných karet vytvářím mapu s 25 místy, v tabulce pěti sloupců a pěti řádků, pomocí relativních souřadnic. Jednoduše zjistím nejmenší a největší souřadnici pro osy X a Y, následně kartu přiřadím do odpovídajícího sektoru. Na jednotlivé místa ukládám možnosti z rozpoznávaných slov. Slovo s největším počtem opakování v daném místě se vybere jako hlavní slovo a dále se tváří jako slovo rozpoznané v této oblasti. Rozpoznávání karet hrací plochy se ve zdrojových souborech nachází ve třídě **Scanner.java**. Získání pozice karty v mřížce lze popsat následujícími vzorci:

$$X = \frac{x_{position} * 5}{x_{max} - x_{min}} \quad Y = \frac{y_{position} * 5}{y_{max} - y_{min}}$$

Obrázek 5.11: Vzorce pro získání pozice v mřížce.



Obrázek 5.12: Obrázek ukazuje nalezení bloku, řádků a nakonec jednotlivých slov. zdroj <https://developers.google.com/vision/android/text-overview>



Obrázek 5.13: Ukázka detekce s využitím relativních pozic při načítání neúplně rozpoznané hrací plochy. Zelenou barvou jsou zobrazeny rozpoznané karty.

5.5 Backend

Aplikace sama o sobě zastává víceméně všechnu potřebnou funkcionalitu, dokáže rozpoznat hrací karty, simulovat hraní a zaznamenávat tahy odpočítat čas. Ovšem nebylo by na škodu rozšířit možnosti aplikace z pouze jednoho herního zařízení a z tohoto důvodu jsem se rozhodl implementovat backendovou část pro sdílení aktuálně rozehrané hry. Běžící server také poslouží jako přístupové místo pro plnění nových náповěd bez nutnosti spouštět aplikaci na mobilním zařízení, ale z pohodlí desktopové platformy.

5.5.1 REST API a JSON

Pro komunikaci se serverem jsem zvolil dnes již standardní architekturu REST, která umožňuje přistupovat ke vzdáleným datům pomocí HTTP metod - viz. [10]. Poskytuje totiž rozhraní pro provádění CRUD operací. CRUD je akronym zahrnující slova Create, Read, Update, Delete. Jedná se o čtyři základní operace nad daty v perzistentním úložišti, které jsou v REST API implementovány pomocí odpovídajících HTTP metod. Get - Retrieve je metoda pro získání dat z konkrétního zdroje, Post - Create je metoda pro vytvoření dat na serveru pomocí předání dat v těle požadavku a zavolání přesně nedefinovaného zdroje. Put - Update je metoda velice podobná metodě Post, s tím rozdílem, že v okamžiku volání již známe konkrétní zdroj, jehož URI specifikujeme v požadavku a také musíme předat data pomocí těla zprávy. Poslední metodou je Delete, která se jmenuje v obou případech stejně a umožňuje smazat zdroj odkazovaný pomocí určité URI podobným voláním jako je GET.

Pro zasílání dat se většinou používá JSON¹⁸, tedy způsob zápisu dat, který je nezávislý na platformě. Jedná se o odlehčenou alternativu k XML. JSON podporuje ukládat čísla, booleovské hodnoty, řetězce, indexované i neindexované pole a objekty, které mají strukturu klíč : hodnota. V případě objektů je klíč uložen jako textový řetězec. JSON umožňuje uložit i speciální hodnotu null.

¹⁸JavaScript Object Notation

5.5.2 Python

K napsání serveru, který by poskytoval tuto funkcionalitu potřebnou pro sdílení her jsem zvolil Python ve verzi 3.6, jedná se o vysokoúrovňový programovací jazyk s dynamickou kontrolou datových typů. Jeden z hlavních důvodů bylo, že pro Python existuje velké množství knihoven a balíčků, které je možno jednoduše přidat do programu pomocí pythonského manažeru balíčků PIP¹⁹. Pro potřeby přijímání a odpovídání na HTTP dotazy lze využít jeden z existujících webových frameworků – buď robustní Django, popřípadě jeden z menších Flask nebo Pyramid. Pomocí PIP jsem tedy nainstaloval Flask²⁰. Práce s balíčkem Flask je velice jednoduchá, k určitým metodám stačí přidat dekorátor `@app.route('/')`, který zaručí, že po přístupu na adresu specifikovanou v dekorátoru se provede právě tato metoda. Jedinou podmínkou je, že metoda musí vrátet hodnotu. Flask podporuje dynamické routování, jedná se o možnost přistupovat k url, která se podobá specifikovanému předpisu, ale předem není celá cesta známá. Konkrétní metoda pro uložení hry pro možnost sdílení na serveru vypadá následovně.

```
# Accepting only POST requests
@app.route('/save/<id>', methods = ['POST'])
def receive(id):
    data = request.get_json(force=True)
    # Set active time to now() + 10 minutes
    saves[id] = {'time':time.time()+600, 'data': data}

    return "Record saved"
```

Na běžícím serveru jsem implementoval i jednoduchou webovou aplikaci, ve které mi uživatelé pomohli naplnit základní sadu nápořád. Sever se proto musí spouštět s parametrem, který udává polohu vstupního souboru. Jeho struktura je pak velice jednoduchá, 400 řádků se slovy oddělenými mezerou, první slovo vždy určuje herní kartu a zbytek nápořád vztahující se k ní. Flask podporuje vykreslování HTML stránek pomocí šablonovacího systému Jinja2. Stačí předat metodě `render_template()` výchozí html soubor a argumenty s libovolným jménem, na které se pak v šabloně pomocí dvojité složené závorky odkazujeme. Jinja podporuje i pokročilejší úkony jako podmínky, cykly, ba dokonce i řazení kolekcí. Ukázka šablony zpracovávané systémem Jinja:

```
<h1>Card: {{card_name}}</h1>
{% for value in hints %}
    <a class="send" id="{{value}}"><span>{{value}}</span></a>
{% endfor %}
```

Server následně umožňuje uložení do struktury shodné se vstupním souborem, ale také přímé vygenerování databázového souboru používaného jako úložiště nápořád v zařízení při hraní pomocí SQLite3. V budoucnu je plánováno i automatické synchronizování všech aktivních zařízení s touto centrální databází, ovšem nyní mne trápí nedostatek času.

¹⁹<https://pypi.org/project/pip>

²⁰<http://flask.pocoo.org>

Codenames Dictionary

Total hints: 5730

Unique: 1236

Average: 14.325

Median: 13.0

Solo hints: 284

Cards

Solos

Duos

Save hints

Save DB

VENUŠE	5	KNÍR	5	KOLEJ	5
HŮL	5	SESTRA	6	ROLE	6
ŽELVA	6	MÍR	6	KLIKA	6
POEZIE	6	MARS	6	MRAVENEK	6
PAS	6	PLOT	6	VÁHA	6
ZADEK	6	TULIPÁN	6	PONOŽKA	6
PAŘÍŽ	6	PLANETA	6	KYVADLO	6
NĚMECKO	7	OBR	7	UCHO	7
ŠROUBEK	7	ZELÍ	7	PARK	7
PLASTELÍNA	7	MUCHOMŮRKA	7	KOSA	7
PRAČKA	7	PÓL	7	JEŽEK	7
PEKING	7	OREL	7	HOLANĎAN	7
OSTROV	7	VÍDEŇ	7	ZOMBIE	7
KANADA	7	VÍČKO	7	RÁDIO	7
VÍRA	7	MOSKVA	7	LONDÝN	7

Obrázek 5.14: Ukázka rozhraní podpůrné webové aplikace pro označování nápověd u karet.

Kapitola 6

Testování a experimenty

Tato kapitola je zaměřená na testování výsledné aplikace, testování je děleno na tři rozdílné části, intuitivnost rozhraní, úspěšnost rozpoznávání a reálný vliv pomoci na výhru týmu. Na závěr kapitoly je rozebráno hodnocení uživatelů. Uživatelské testování proběhlo dne 6.5. večer a 7.5. ráno, účastnilo se jej celkem 10 uživatelů. Bohužel hned prvním problémem v jednoduchosti přístupu k aplikaci je její momentální absence v obchodě Google Play. Sestavený aplikační balíček byl tedy nejdříve rozeslán pomocí jedné ze světových komunikačních služeb a po povolení instalace ze zdrojů třetích stran v zařízení byla aplikace nainstalována celkem na 5 zařízení s operačním systémem Android ve verzi 5.0 nebo vyšší. Aplikaci nebylo možno nainstalovat na všech deset zařízení, protože 3 účastníci vlastní telefon Apple iPhone a jeden z účastníků měl Android v nižší verzi.

6.1 Intuitivnost rozhraní

Zjišťování velice subjektivního dojmu, jak moc je rozhraní intuitivní, jsem prováděl zadáním několika úkolů uživatelům. Na počátku testování ležela na stole s bílým ubrusem rozložená hra Krycí jména, každý z účastníku dostal psací potřeby a kus papíru pro případné poznámky. Všechna zařízení byla připojená k domácí WiFi síti pro otestování funkcí jako je sdílení, a ve vedlejší řadě i ověření funkčnosti při použití serverů třetích stran. Před začátkem jsem se ještě ujistil, zda jsou pravidla hry všem přítomným známa, a pro jistotu jsem je ještě ve zkrácené verzi zopakoval.

6.1.1 Objevování aplikace

Prvním úkolem bylo, aby si uživatel aplikaci sám proklikal, zjistil k čemu slouží různá tlačítka, objevil co nejvíce možných funkcí a až ho průzkum přestane bavit, aby beze slov propůjčil mobilní zařízení některému z kolegů, který aplikaci ještě nevyzkoušel. Téměř všichni uživatelé byli s průzkumem aplikace do 4 minut hotoví. Ranní běh na tom byl časově lépe oproti večernímu, jak se ukázalo, mohla za to bariéra před spuštěním hry související se skenováním. Po ukončení testování všemi uživateli jsem se ptal na nalezené funkce. Všichni zúčastnění našli v hlavním menu nastavení a možnost načtení hry, které v prvním momentu spuštění aplikace bylo zcela prázdné. Všichni také pochopili, kdy po nich aplikace vyžaduje skenování hrací plochy nebo herního plánu.

V pohledu hlavního agenta našli funkci nápovědy. Na záložce Record se bohužel většina uživatelů ztratila, tlačítka po zmáčknutí neprovádí žádnou akci a uživatel nevěděl co dále. Pouze dva lidé zjistili, že po podržení prstu na tlačítko začne měnit tvar. Zobrazení

Jméno	Pohlaví	Věk	Vlastní smartphone	Zařízení
Jakub	Muž	25	9 let	—
David	Muž	44	5 let	Xiaomi Redmi Note 4
Veronika	Žena	23	7 let	Samsung Galaxy J5
Petra	Žena	44	3 roky	Xiaomi Redmi 4X
Roman	Muž	28	4 roky	—
Marcela	Žena	36	2 roky	—
Petr	Muž	20	5 let	Huawei Y6
Jakub	Muž	21	8 let	—
Kamila	Žena	23	6 let	Xiaomi Redmi Note 3
Veronika	Žena	20	3 roky	Nokia 6 Matte

Tabulka 6.1: Záznam jednotlivých sérií, prostřední sloupec udává počet správně a chybně načtených karet na jednotlivých zařízeních, pravý sloupec popisuje důvody, proč nebyla hrací plocha načtena korektně.

Zařízení	Verze OS	Displej	Rozlišení	Fotoaparát
Xiaomi Redmi Note 4	7.0	5.5"	1920 x 1080	13 Mpix
Xiaomi Redmi 4X	6.0	5"	1280 x 720	13 Mpix
Samsung Galaxy J5	7.0	5"	1280 x 720	13 Mpix
Huawei Y6	5.1	5.2"	1280 x 720	8 Mpix
Xiaomi Redmi Note 3	6.0	5.5"	1920 x 1080	13 Mpix
Nokia 6 Matte	7.1	5.5"	1920 x 1080	16 Mpix

Tabulka 6.2: Seznam zařízení, na kterých byla aplikace testována. S výkonem aplikace problémy nebyly a proto parametry související s ním neuvádím.

dialogového okna tahem po displeji pro zadání nápovědy nenašel nikdo. Další dvě záložky byly všem uživatelům zcela srozumitelné. Většina dokonce vyzkoušela i odpočet času, sdílení a následné načtení cizí hry. Zpozorovali i možnost uložení záznamu .CNR, ovšem již údajně neprohledávali úložiště zařízení pro možnost otevření záznamu.

Z pohledu operativce zjistili uživatelé možnost označování karet, většina našla i označení karty jako zamýšlené, ovšem nevěděli co daná symbolika znamená. Na záložce Hints byli schopni přidávat nápovědy pomocí dobře viditelného tlačítka otevírající dialogové okno a následně pozorovat měnící se stav na první záložce – Cards. Odpočet času a sdílení hry jim již bylo známé z minulé obrazovky.

Zeptal jsem se na konfiguraci pomocí dialogu nastavení v hlavním menu. Nikdo z přítomných neměl problém změnit přezdívkou, do úpravy IP adres serverů se nepouštěli, většina vyzkoušela změnu přepínačů pro pokročilé možnosti nápověd a využití serverů, ale před pokračováním je vrátili zase do původní polohy. Znovu se, jak jsem zjistil, do nastavení již nikdo nevrátil.

6.2 Úspěšnost rozpoznávání

Úspěšnost rozpoznávání je kategorie, která na rozdíl od intuitivnosti aplikace nesouvisí s chápáním uživatele, ale je kompletně v režii technologií mobilu a napsaném algoritmu. Samotné rozpoznávání je sice zajišťováno třetí stranou, knihovnou od společnosti Google, ta ovšem mezi podporované jazyky nezařazuje češtinu, proto se musím starat o dodatečnou korekci a rozhodl jsem se zařadit tuto část mezi předměty testování. Po správném odhalení slov ještě nastává otázka správného rozřazení slov do sloupců.

6.2.1 Hrací plocha

Postupně se na stůl vyskládalo 6 sérií karet, které byli uživatelé nuceni naskenovat. Následně jsem v každé sérii provedl pár (4-7) změn, které se týkaly otočení karty a tedy změny textu, nebo změny pozice karty (prohození dvou a více karet). Celkem tedy uživatelé provedli 60 načtení hrací plochy ve dvou bězích (30 večer a dalších 30 následující ráno). Každé načtení jsem pečlivě zaznačil a poznamenal si také hodnoty související převážně s chybnou detekcí textu. Uživatelé, kterým se detekce na první pokus nepodařila jsem požádal, aby se vrátili do hlavního menu a detekci provedli znovu.

6.2.2 Herní plán

Druhá část, potřebná ke hraní hry, která souvisí s rozpoznáváním je načítání herního plánu. Na stůl jsem vyskládal všech 40 herních plánů a poprosil přítomné o naskenování aspoň šesti kusů a případné prodiskutování nesrovnalostí.

6.3 Vliv na výhru

Poslední a časově nejvíce náročná část bylo zhodnocení jak moc je nápověda opravdu prospěšná a jak velkým dílem ovlivní šance na výhru soupeřících týmů. Z balíčku všech karet jsem vyčlenil 100 náhodných kusů, celkem tedy 200 slov odpovídajících osmi hrám a utvořily se týmy po šesti hráčích včetně mne. Poté začalo samotné hraní s přesně danými poměry, který tým, kdy a jak mohl využívat pomoci aplikace. Začínající tým s úkolem uhádnout 9 karet je dále uváděn jako tým 1, týmu hrajícímu jako druhému v pořadí se přezdívá

Série	Správných načtení			Hlavní chyby
	Z1	Z2	Z3	
1.	24	25	24	Diakritika: Karta plášť zaměněna za plast
2.	25	25	25	—
3.	25	25	25	—
4.	25	23	25	Duplikace karty na špatné pozici
5.	23	25	24	Nerozpoznání karet: (KARBANÍK , RUČNÍK), (RUČNÍK)
6.	25	25	24	Nerozpoznání karet: (RUČNÍK)
7.	25	25	25	—
8.	24	25	25	Nerozpoznání karet: (TLAČÍTKO)
9.	25	25	25	—
10.	25	24	25	Diakritika: Karta plášť zaměněna za plast
11.	25	25	24	Duplikace karty na špatné pozici
12.	25	25	25	—

Tabulka 6.3: Záznam jednotlivých sérií, prostřední sloupec udává počet správně načtených karet (z maximálního počtu 25) na jednotlivých zařízeních (Z1 až Z3), pravý sloupec popisuje důvody, proč nebyla hrací plocha načtena korektně.

tým 2. Konfigurace aplikace zahrnovala zapnutou možnost pokročilých návodů pro hlavního agenta i operativce. Ve večerním běhu vypadaly hry následovně.

- **První hra** - Tým 1 využíval pomoc pouze pro hlavního agenta, operativci ani soupeř aplikaci používat nemohli.
Předpokládaný scénář: Výhra týmu 1, tým 1 neoznačí ani jednoho náhodného kolemdoucího nebo nájemného vraha.
- **Druhá hra** - Tým 1 využíval pomoc pouze pro operativce, hlavní agent ani tým 2 aplikaci používat nemohli.
Předpokládaný scénář: Výhra týmu 1.
- **Třetí hra** - Tým 1 využíval pomoc pro hlavního agenta i operativce, tým 2 aplikaci používat nemohli.
Předpokládaný scénář: Výhra týmu 1, tým 1 neoznačí ani jednoho náhodného kolemdoucího nebo nájemného vraha.
- **Čtvrtá hra** - Oba týmy využívaly pomoc pouze pro hlavního agenta, operativci aplikaci používat nemohli.
Předpokládaný scénář: Výhra týmu, který neoznačí ani jednoho náhodného kolemdoucího nebo nájemného vraha.
- **Pátá hra** - Tým 1 využíval pomoc pro hlavního agenta, operativci aplikaci používat nemohli a tým 2 využíval pomoc pouze pro operativce.
Předpokládaný scénář: Výhra týmu 2, který může vyloučit karty prvního týmu.
- **Šestá hra** - Tým 1 využíval pomoc pro hlavního agenta i operativce, tým 2 využíval pomoc pouze pro operativce.
Předpokládaný scénář: Výhra týmu 1.
- **Sedmá hra** - Oba týmy využívaly pomoc pro hlavního agenta i operativce.
Předpokládaný scénář: Výhra začínajícího týmu (týmu 1).

- **Osmá hra** - Tým 1 využíval pomoc pouze pro operativce, tým 2 využíval pomoc pro hlavního agenta i operativce.

Předpokládaný scénář: Výhra týmu 2.

Zaznamenané výsledky z večerního běhu je možno vidět v tabulce 6.4, z ranního běhu pak v tabulce 6.5.

Hra	Výsledné skóre		Kolemjdoucích		Příčina konce
	1 Tým	2. Tým	1 Tým	2. Tým	
První	9	7	0	1	Výhra týmu 1
Druhá	8	8	1	1	Výhra týmu 2
Třetí	3	1	0	0	Tým 2 označil vraha
Čtvrtá	7	8	1	0	Výhra týmu 2
Pátá	7	8	0	1	Výhra týmu 2
Šestá	9	7	0	1	Výhra týmu 1
Sedmá	9	7	0	0	Výhra týmu 1
Osmá	8	8	1	0	Výhra týmu 2

Tabulka 6.4: Záznam z her večerního běhu, první dva sloupce určují kolik karet tým uhádnul, druhé dva kolik označil náhodných kolemjdoucích.

V ranním běhu vypadaly hry skoro úplně totožně, jediným rozdílem bylo zaměnění možností u týmu 1 a týmu 2. Ke hraní posloužila zbývající sada doposud nepoužitých 100 karet.

Hra	Výsledné skóre		Kolemjdoucích		Příčina konce
	1 Tým	2. Tým	1 Tým	2. Tým	
První	8	8	1	0	Výhra týmu 2
Druhá	8	8	0	1	Výhra týmu 2
Třetí	6	8	2	0	Výhra týmu 2
Čtvrtá	9	7	0	1	Výhra týmu 1
Pátá	9	6	1	1	Výhra týmu 1
Šestá	7	8	0	0	Výhra týmu 2
Sedmá	9	7	0	0	Výhra týmu 1
Osmá	5	4	0	0	Tým 2 označil vraha

Tabulka 6.5: Záznam z her ranního běhu, první dva sloupce určují kolik karet tým uhádnul, druhé dva kolik označil náhodných kolemjdoucích.

6.4 Vyhodnocení

6.4.1 Rozhraní

Jak se ukázalo, v rozhraní aplikace jsou mezery, které na uživatele nemající přehled o všech přítomných funkcích působí tak, že funkce se v aplikaci ani nenachází. Nejrazantnější ukázkou tohoto chování bylo přidávání nápověd v pohledu hlavního agenta, což uznávám, protože k potažení prstem po displeji většinou nemáme ve stávajících aplikacích důvod. Potýkal jsem se ale s nedostatkem místa a proto jsem musel použít toto řešení. Nyní se mi zdá, že lepší přístup k testování mohlo být sepsání všech přítomných funkcí aplikace na tabuli a

požádat uživatele, aby je všechny našel. Zde by se mohla projevit jistá vazba s obrazovkou v pohledu operativce, kde v druhé záložce je možno přidávat nápovědy, uživatel by tak s větší šancí náhodnými gesty dialogové okno nakonec vypátral. Každopádně na základě výsledků této části testu, jsem se rozhodl implementovat nápovědu, která se uživateli zobrazí při prvním spuštění aplikace, popřípadě ji bude možno vyvolat z hlavního menu.

Ne přímo potíže, měli uživatelé s obrazovkou pro korekci případně ručí zadávání karet, ale tato obrazovka, v případě kdy se nepovedla stoprocentní detekce, neshledala velké ovace. Problémem bylo zejména zdlouhavé zadávání karet a potřeba provést několik úkonů. Bohužel tento proces by šlo zjednodušit, pokud by se například samy nabízely slova v průběhu kdy uživatel zadává text do aplikace, na to ale nezbyl čas. Toto vylepšení by se v aplikaci mohlo do budoucna objevit.

6.4.2 Rozpoznání hracích karet

Z celkového počtu 36 načtení bylo 27 korektních načtení na první pokus. Přitom jen ve dvou případech se počet chybných karet dostal na hranici dvou kusů. Dle mého názoru je toto kladný výsledek. Jak jsem předpokládal, problém dělaly převážně karty obsahující v názvu diakritiku. Řešení tohoto problému by mohlo být akceptovat odchylky a přiřazovat jim nejbližší možné slovo. Vypracování této metody by nebylo až tak pracné, jako spíše časově náročné. Potřeboval bych také zjistit, jestli tento způsob nezapříčiní více překlepu než v aktuálním stavu a proto jsem s odchylkami nepočítal.

Případy kdy byla karta zduplikována na jiné pozici, než se nachází je bohužel chyba algoritmu, který v reálném čase určuje pozici karet. Málokdy se ale podaří knihovně rozpoznat všech 25 názvů najednou v jednom snímku přicházejícího z fotoaparátu a mapa slov se tvoří postupně. Aby nedocházelo k zaplnění paměti a pádu aplikace, přestává se mapa tvořit na místech kde se vyskytuje již dostatečný počet možností. Nejjednodušší řešení je v tomto případě znovu otevřít skenování herní plochy v okamžiku, kdy fotoaparát již míří na samotné hrací karty.

6.4.3 Rozpoznání herního plánu

Po dobu načítání herních plánů si nikdo nestěžoval na chybné načtení. Načítání si vedlo až překvapivě dobře, se stoprocentní úspěšností, a proto jsem později provedl ještě sérii pokusů, za nedokonalého osvětlení a na jiné podložce než byl bílý ubrus. Zde již byly výsledky horší hlavně co se trvání týče, ovšem vždy jsem byl schopný po natočení se za světlem, nakloněním případně přiložením karty na své tělo herní plán načíst.

Kapitola 7

Závěr

Cílem této práce bylo vytvořit aplikaci, která jakýmkoliv způsobem pomůže uživateli při hraní hry Krycí jména, tento požadavek jsem si vyložil jako zvýšení šancí na výhru a v tomto duchu jsem celou aplikaci implementoval. Přitom jsem se snažil, aby složitým rozhraním nebo dlouhými prostoji při přípravě ke hraní aplikace uživatele neodradila od používání. Téměř stoprocentní načítání karet se provede v řádu okamžiků a poté aplikace ukáže svůj plný potenciál v podobě uživatelského pomocníka. Zároveň bych si dovilil zdůraznit, že aplikace v základní podobě umožňuje spuštění i hraní bez připojení k internetu. Stává se tak skvělým společníkem ke každé příležitosti hraní Krycích jmen. Samotná aplikace se soustředí hlavně na operace nad neustále se rozšiřující sadou slov a s každým zaznamenaným tahem radí uživateli jakým způsobem se nejvíce přiblížit vítězství. Na rozdíl od podobných svého druhu, se nejedná pouze o jakési rozšíření hrací plochy nebo herních prvků na obrazovky mobilů.

Jedno z možných rozšíření, nad kterým jsem během vývoje přemýšlel by mohla být synchronizace celé hry v reálném čase, tedy sdílení na všechna zařízení, která se ke hře připojí. Zadávání kroků ve hře by se tak rozdělilo mezi několik hráčů. Dalším rozšířením, ale o něco jednodušším na implementaci ovšem celkem časově náročným by se mohla stát podpora cizích jazyků oproti pouhé češtině. Aplikaci by bylo ideální napojit na data získané například v práci Viliama Samuela Hošťáka – *Shlukování slov podle významu*[5], podobná funkcionality se zde již částečně vyskytuje.

Literatura

- [1] *SQLite Documentation*. [Online; navštíveno 23.03.2018].
URL <https://www.sqlite.org/docs.html>
- [2] Cabrera, B.: *HSV*. [Online; navštíveno 01.05.2018].
URL <http://obsessive-coffee-disorder.com/tag/hsv/>
- [3] Developers, A.: *Documentation for app developers*. [Online; navštíveno 13.11.2017].
URL <https://developer.android.com/docs/>
- [4] Dorson, R.; Golub, E.: *CrossScan: The Crossword Scanning App*. online, 2014.
- [5] Hošták, V. S.: *Shlukování slov podle významu*. 2016.
- [6] Hrbas, V.: *Řešitel sudoku pro Android*. Bakalářská práce, Vysoké učení technické v Brně, Fakulta informačních technologií, 2014.
URL <http://www.fit.vutbr.cz/study/DP/BP.php?id=16379>
- [7] Krčmář, M.: *Zpracování obrazu v zařízení Android - detekce a rozpoznání vizitky*. 2016.
URL <http://hdl.handle.net/11012/58795>
- [8] Luboslav Lacko: *Vývoj aplikací pro Android*. Brno: Computer Press, první vydání, 2015, ISBN 978-80-251-4347-6.
- [9] Laganière, R.: *OpenCV 2 computer vision application programming cookbook*. Brimingham: Packt Publishing, první vydání, 2011, ISBN 978-1-84951-324-1.
- [10] Malý, M.: *REST: architektura pro webové API*. [Online; navštíveno 18.04.2018].
URL <https://www.zdrojak.cz/clanky/rest-architektura-pro-webove-api>
- [11] Mednieks, Z. R.: *Programming Android*. Sebastopol: O'Reilly, první vydání, 2011, ISBN 978-1-449-38969-7.
- [12] Šonka, M.; Hlaváč, V.: *Počítačové vidění*. Praha: Grada, 1992, ISBN 80-85424-67-3.

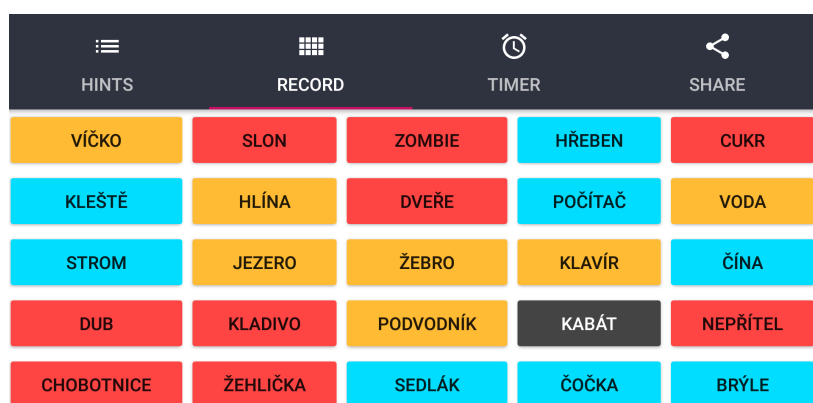
Příloha A

Sada snímků v různých fázích hry

V této části přílohy se nachází několik dvě sady snímků pořízených během hraní hry Krycí jména. Snímky až na výjimku tvoří dvojice, aktuální stav na stole a k němu korespondující zobrazení na obrazovce v pohledu hlavního agenta. Pouze na obrázku A.5 je zobrazena obrazovka v pohledu operativce, po několika zadaných nápovědách.



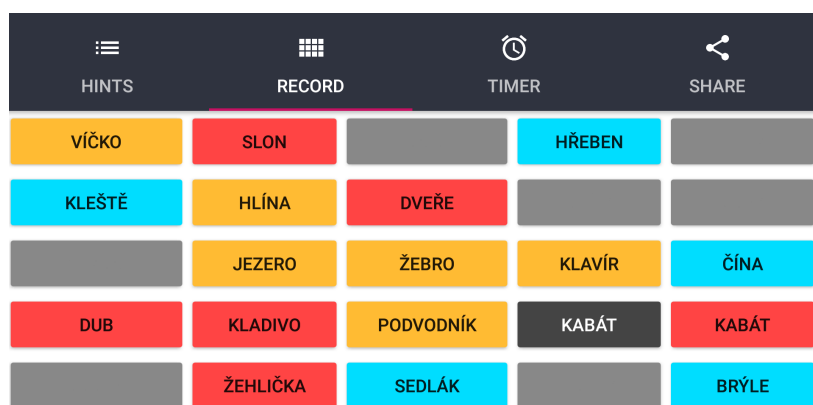
Obrázek A.1: Rozložení herní plochy a stav na začátku hry : Stav hracích karet.





































Obrázek A.2: Rozložení herní plochy a stav na začátku hry : Vyobrazení v elektronické podobě.



Obrázek A.3: Rozložení herní plochy a stav v průběhu hry : Stav hracích karet.



Obrázek A.4: Rozložení herní plochy a stav v průběhu hry : Vyobrazení v elektronické podobě.

 CARDS	 HINTS	 TIMER	 SHARE			
 HŮL		1.750000				
 KEŘ		1.000000				
 OHEŇ		1.000000				
 HLINÍK		0.750000				
 ŘETĚZ		0.750000				
 ANDĚL		0.000000				

Obrázek A.5: Rozložení herní plochy a stav na konci hry: Vyobrazení v elektronické podobě z pohledu operativce.



Obrázek A.6: Rozložení herní plochy a stav na konci hry : Stav hracích karet.

HINTS	RECORD	TIMER	SHARE
KLEŠTĚ	HLÍNA		
		ŽEBRO	KLAVÍR
		PODVODNÍK	KABÁT
		SEDLÁK	

Obrázek A.7: Rozložení herní plochy a stav na konci hry: Vyobrazení v elektronické podobě.

Příloha B

Návod pro spuštění podpůrného serveru

V této příloze je popsána instalace potřebného balíku a spouštění podpůrného serveru na zařízení, kde je možné spustit Python ve verzi 3.6. Jediný balík potřebný k doinstalování je Flask.

Instalace balíku Flask:

```
sudo pip install Flask
```

Formát spuštění:

```
python server.py <inputfile1> [<inputfile2> <inputfile3> ... ]
```

- **<inputfile1>**: Jméno vstupního souboru s délkou 400 řádků. Nachází se na přiloženém médiu
- **<inputfile2> <inputfile3> ...** : Pokud je potřeba, aplikace dokáže načíst libovolné množství vstupních souborů stejného formátu

Příklad spuštění:

```
python server.py input.txt
```

Nyní by měl server oznámit, že naslouchá na portu 5000. Webová aplikace běží na adrese <http://127.0.0.1:5000/fill>.